**Lógica de la aplicación**

El proyecto de Java está hecho siguiendo el paradigma de Programación Orientada a Objetos (POO). Esta cuenta con dos clases:

* Main: la cual es la clase que contiene el método main y por la cual la aplicación se ejecuta.
* Persona: la cual tiene los atributos y métodos necesarios que la aplicación necesita.

**Clase Persona**

En la clase Persona indicamos los atributos que se necesitan para registrar una persona (nombre, apellido, DNI y fecha de nacimiento), además de un atributo estático que funciona como id de la persona. Estos atributos se encuentran encapsulados con el modificador de acceso “private”.

También encontramos el constructor el cual recibe tiene como argumentos el nombre, apellido, DNI y fecha de nacimiento que luego le pasaremos para poder instanciar el objeto de tipo Persona. Por último, cuenta el método “toString”, el cual está sobrescrito para poder mostrar la información de la persona/objeto.

**Clase Main**

En la clase Main declaramos las variables de tipo String nombre, apellido, DNI y fechaNacimiento y instanciamos un ArrayList llamado listaPersonas, en el cual guardaremos las personas registradas.

La aplicación comienza mostrando un menú principal con las opciones: registrar persona, mostrar personas y salir. Este menú se encuentra dentro de un ciclo Do-While, donde se declara un método Scanner y una variable “opc” la cual indica cual de las 3 opciones queremos ejecutar. La condición del while es si “opc” es diferente a 0, esto para que, si se ingresa otro número que no sea ni 1, 2 o 0, la aplicación le indique que tienen que volver a elegir una opción que esté dentro del menú.

Luego, con un condicional Switch controlamos los casos que el usuario ingrese: caso 1 para registrar personas, caso 2 para mostrar la lista y caso 0 para salir del programa. Además de un default el cual nos indica que ingresemos nuevamente una de las opciones anteriores, en caso de que se digite otro número.

Caso 1: Se pide al usuario que ingrese los datos para registrar a la persona.

* Se pide que ingrese el nombre, el cual el valor de la entrada (método Scanner) se guarda en la variable “nombre”. La validación que se utiliza es while donde la condición es que, si el nombre es igual a un String vacío, debe volver a ingresar el nombre. El while no finaliza hasta que el valor sea diferente a un string vacío.
* Se pide que ingrese el apellido y se guarda en la variable “apellido”. Se utiliza la misma validación que nombre, solo que ahora se usa la variable apellido.
* Se pide que ingrese un DNI de 8 dígitos, el valor se guarda en la variable “DNI”. En la validación también se utiliza un while donde la condición es que, si la longitud del String DNI es diferente a 8, debe ingresar un DNI válido. El while termina cuando se ingresa un string de 8 dígitos.
* Por ultimo se pide la fecha de nacimiento y esta se guarda en la variable “fechaNacimiento”. En la validación también se utiliza un ciclo while donde mientras la condición de que la fecha coincida con la expresión regular sea falsa, se pide al usuario que ingrese la fecha correcta siguiendo la forma de dd/mm/yyyy.

Una vez que se ingresaron todos los datos, se instancia un objeto de tipo Persona y se le pasa como argumento y utilizando el constructor de la clase Persona, las variables que contienen los datos que el usuario ingreso. Por último, se añade al objeto al ArrayList.

Caso 2: El usuario desea ver la lista con las personas registradas. Se utiliza un if donde la condición es que, si la lista está vacía se muestra un mensaje por consola diciendo que aún no se han registrado personas, de lo contrario y utilizando un else, se recorre el ArrayList. Para recorrer el ArrayList se utiliza un ForEach, el cual va recorriendo los elementos del ArrayList.

Caso 0: El usuario elige salir del programa y se muestra un mensaje por consola agradeciendo al usuario y muestra el nombre del desarrollador. De esta forma, la ejecución de la aplicación finaliza.