

“Actividad 2”

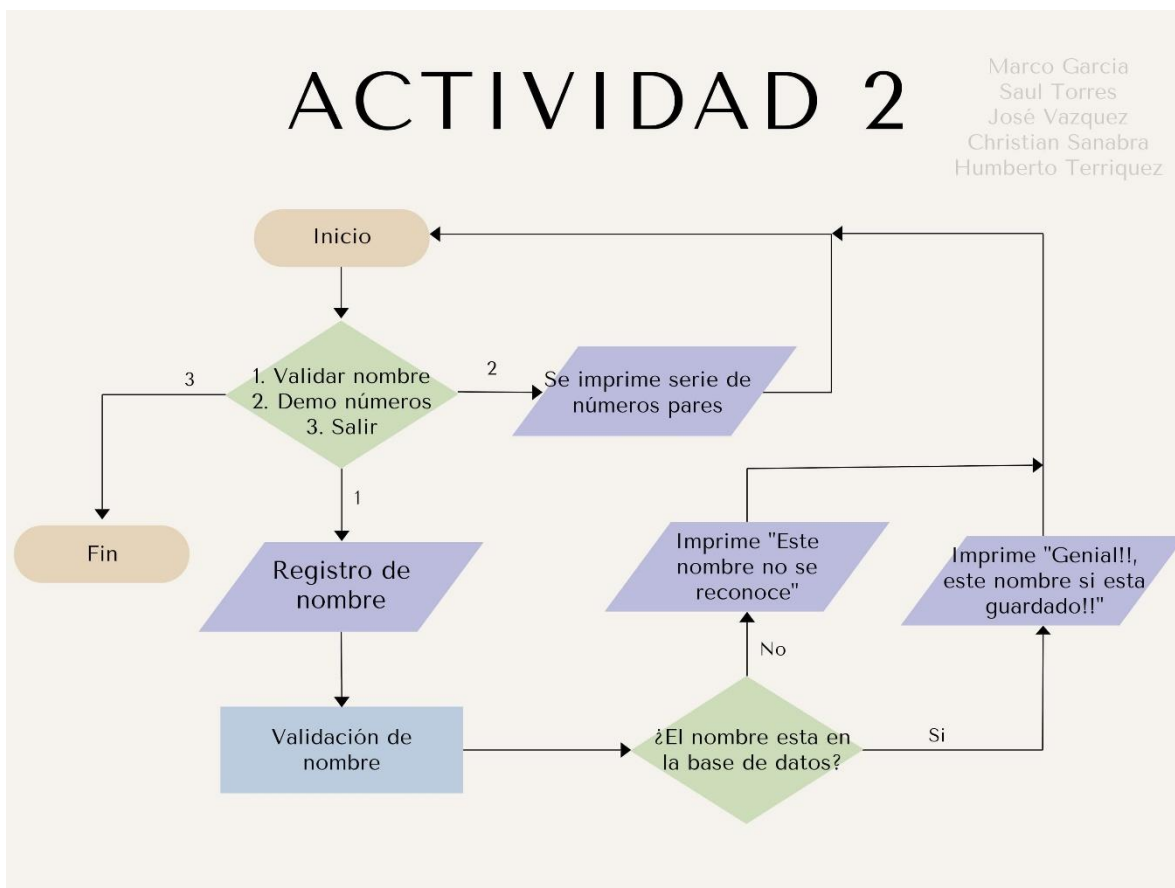
Programación Orientada a Objetos

“Profesor, Carlos Iván Castillo Sepúlveda”

José Antonio Vázquez Mora

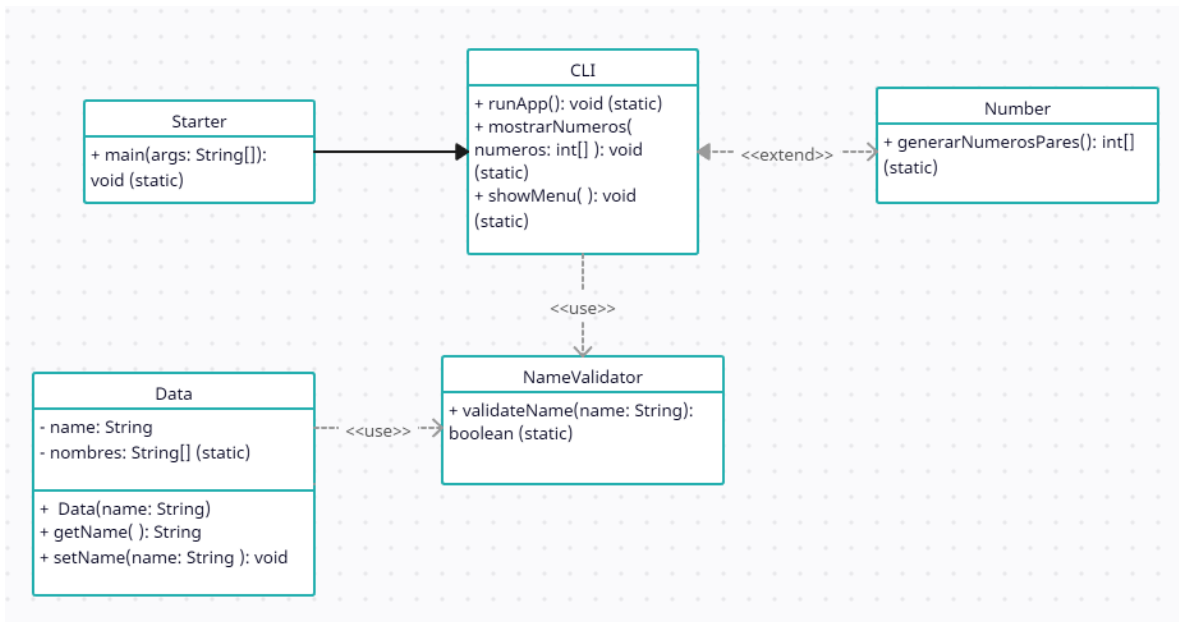
A107087360

En el presente documento resolvimos un ejercicio presentado por nuestro profesor donde tenemos que presentar una lista con los números pares que hay entre el 2 y el 100, de igual manera también tenemos una lista con nombres aprobados y se solicitará que el usuario nos de un nombre para compararlo con la lista, en caso de ser el mismo nombre que la lista nos mostrará un mensaje de coincidencia, por otro lado, en caso de que el nombre no se encuentre en la lista también mostrará un mensaje de error.



En el diagrama de flujo podemos observar el funcionamiento de nuestro programa, al momento de iniciar se muestra un menú al usuario con 3 opciones, validar el nombre, demo de números pares y salir, en caso de elegir la opción 1 nos solicitará un nombre en terminal, después de esto nuestro programa hará una comparación entre el nombre dado y

los nombres que se encuentran en la lista de aprobados, en caso de que los nombres coincidan se imprimirá un mensaje de reconocimiento o desconocimiento según sea el caso, por otro lado, al momento de presionar la opción 2 vamos a poder mostrar en pantalla los números pares entre el 2 y 100 en una lista ante el usuario, por último, en caso de querer salir del menú solo necesita utilizar la opción 3 para terminar con el programa.



Este es el diagrama de clases que utilizamos para trabajar con nuestro programa, como se puede observar Starter es la clase desde la cual vamos a correr nuestra aplicación, en la clase CLI tenemos todo lo relacionado con los inputs del usuario y la manera en la que el programa se va a comunicar con el usuario, con el método `showMenu` es que vamos a mostrar nuestro menú en terminal, con `mostrarNumeros` tenemos el método que manda a llamar la lista de los números pares entre el 2 y el 100 como se nos estaba solicitando, por último `runApp` engloba todos estos procesos para poder inicializarlo posteriormente en Starter. En nuestra clase **Number** tenemos el método `generarNumeros`, que es el encargado de generar la lista de 50 espacios para posteriormente solo almacenar los números pares,

por otro lado, en NameValidator vamos a realizar todo el proceso lógico para encontrar coincidencias entre el nombre que nos proporciona el usuario y una lista previamente definida en la clase Data, al momento de hacer esta comparación la clase nameValidator nos notificará si existe una coincidencia o si no la hay, dando esto un mensaje al usuario final en el terminal.

Podemos concluir con este programa que ya contamos con los conocimientos para trabajar y consultar datos en listas que previamente definimos o que podríamos solicitarle al usuario en un futuro, creo que pudimos aprender bastante de como se comportan los algoritmos al momento de trabajar con diferentes iteraciones ya que no funciona como en casos anteriores donde la consulta es con un único valor, en este caso la consulta es 1 por 1 a lo largo de una lista que podría ser de una cantidad infinita de argumentos. También creo que pudimos aprender a almacenar información y separar lo que realmente queremos como fue el caso en el ejercicio de los números pares entre el 2 y 100, si conocemos la lógica detrás de lo que tenemos que hacer creo que podemos entender mucho mejor al momento de tener un problema él como resolverlo.