

# VetAnimalData

Juan Andrés Barrera Rodríguez, Sara Valentina  
Cardona Mejía, Iván Alexander Morales Muñoz,  
Nelson David Ramírez Marín, Santiago Dleon  
Sánchez Romero



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA  
SEDE BOGOTÁ



## Problema a resolver

El problema que pueden llegar a presentar diversos establecimientos de rescate y adopción radica en que no se cuenta con buen manejo de información sobre los diferentes animales, en este caso gatos y perros, que llegan y que se van del establecimiento, por eso con el proyecto se propone llegar a una herramienta práctica y eficaz que logre tener toda la información sobre los diversos animales que se encuentran en el lugar.



Historia Clínica 0365334

Fecha: 14/03/2018

Nombre de la mascota: Huente

Especie: Perro

Raza: Mixto

Sexo: ♀

Fecha nac.: 26-11-2018

Pelaje: rojo y negro

Tamaño: .....

Observaciones: .....

Propietario: .....

Dirección: .....

Publicaciones

Filtrar CONFIRMAR

Publicaciones cercanas ☒

Radio: 30 km

Tipo publicación: Adopción

Animal: Perro, Gato

Tamaño: Pequeño Grande

Edad: 1 mes 1 año

ADOPTIÓN

Home Star Paw Gear



## Requerimientos funcionales

*Nombre de la funcionalidad:* Creación de Hoja de vida de un animal

*Requerimientos funcionales:* Asegurar que el dato ingresado del animal, corresponda al solicitado. Los datos por cada animal son: nombre, si es perro o gato, género, la fecha de ingreso, si es adoptable, la fecha de salida si este ha sido adoptado y el estado de ingreso. En este último dato los posibles valores son: Muy malo, malo, bueno y muy bueno.

La hoja de vida del animal se almacena en la base de datos.

*Nombre de la funcionalidad:* Actualización Hoja de vida de un animal

*Requerimientos funcionales:* Se necesita mostrar al usuario la información asociada al animal al que se le quiere actualizar su información. El software comprueba que el nuevo valor corresponde al tipo del que se quiere actualizar. En caso de que no, el software NO debe permitir la modificación con este nuevo valor.



## Requerimientos funcionales

*Nombre de la funcionalidad:* Eliminación de hoja de vida de animales adoptados

*Requerimientos funcionales:* Confirmar con un sí o un no sobre la adopción de un animal, y además confirmar de la misma manera si se desea eliminar la hoja de vida con el nombre del animal. Si se colocan entradas diferentes a las expuestas anteriormente el software dará un mensaje de error con la frase “entrada no válida”.

*Nombre de la funcionalidad:* Búsqueda y filtro de hojas de vida de animales

*Requerimientos funcionales:* Se necesita preguntar al usuario si este desea aplicar filtros. Si es el caso, el usuario podrá aplicar los siguientes filtros:

- Si es perro o gato
- Si es adoptable o no

Los filtros se pueden usar en conjunto, o uno sin el otro. En caso de que el usuario no necesite una búsqueda parcial, simplemente se muestran todas las hojas de vida de animales.



## **Uso de estructuras de datos en la solución del problema a resolver**

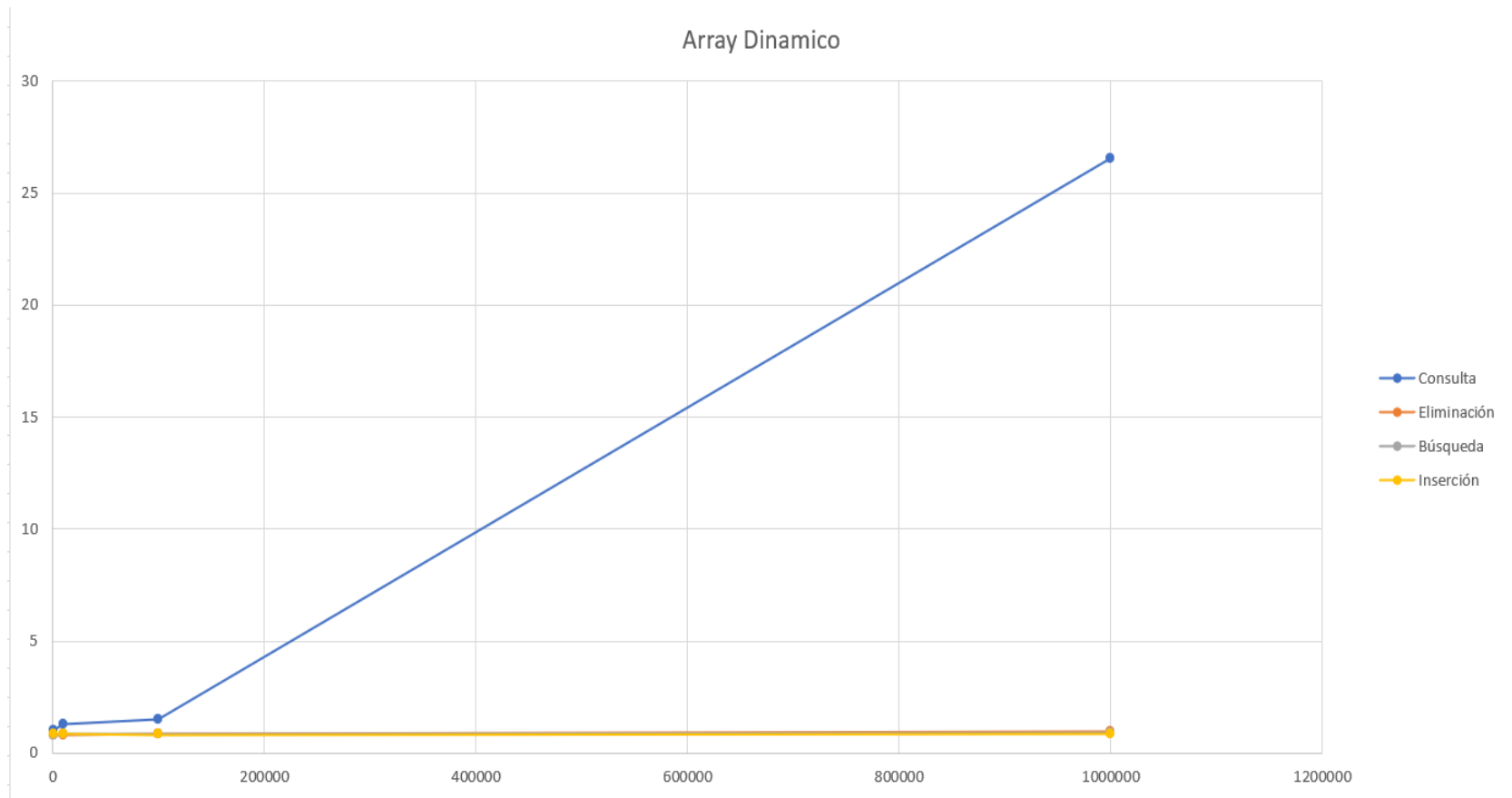
Para el desarrollo del proyecto, se realizó un análisis de las funcionalidades que este debía tener para cumplir con los objetivos propuestos, a partir de esto se concluye que las estructuras de datos que cumplían mejor estos requerimientos serán implementadas en el primer prototipo del proyecto, dichas estructuras fueron arreglos dinámicos, listas enlazadas y pilas. Estas estructuras fueron empleadas para almacenar registros de los animales que han ingresado a la fundación, esta información comprende datos específicos del ingreso del animal como fecha de ingreso, fecha de salida, género, estado de ingreso y condición de adopción.

Cada una de las estructuras usadas almacenan la misma información, esto se hizo con el fin de comparar teóricamente la complejidad de cada acción que se realiza en ellas. Una vez se establecieron las estructuras que serían implementadas, fue necesario crear las funcionalidades que cada una de ellas tendría. Finalmente los métodos implementados fueron ingreso, búsqueda y eliminación de datos, adicionalmente se incluyó una acción que permite mostrar en una tabla los datos almacenados en la estructura.

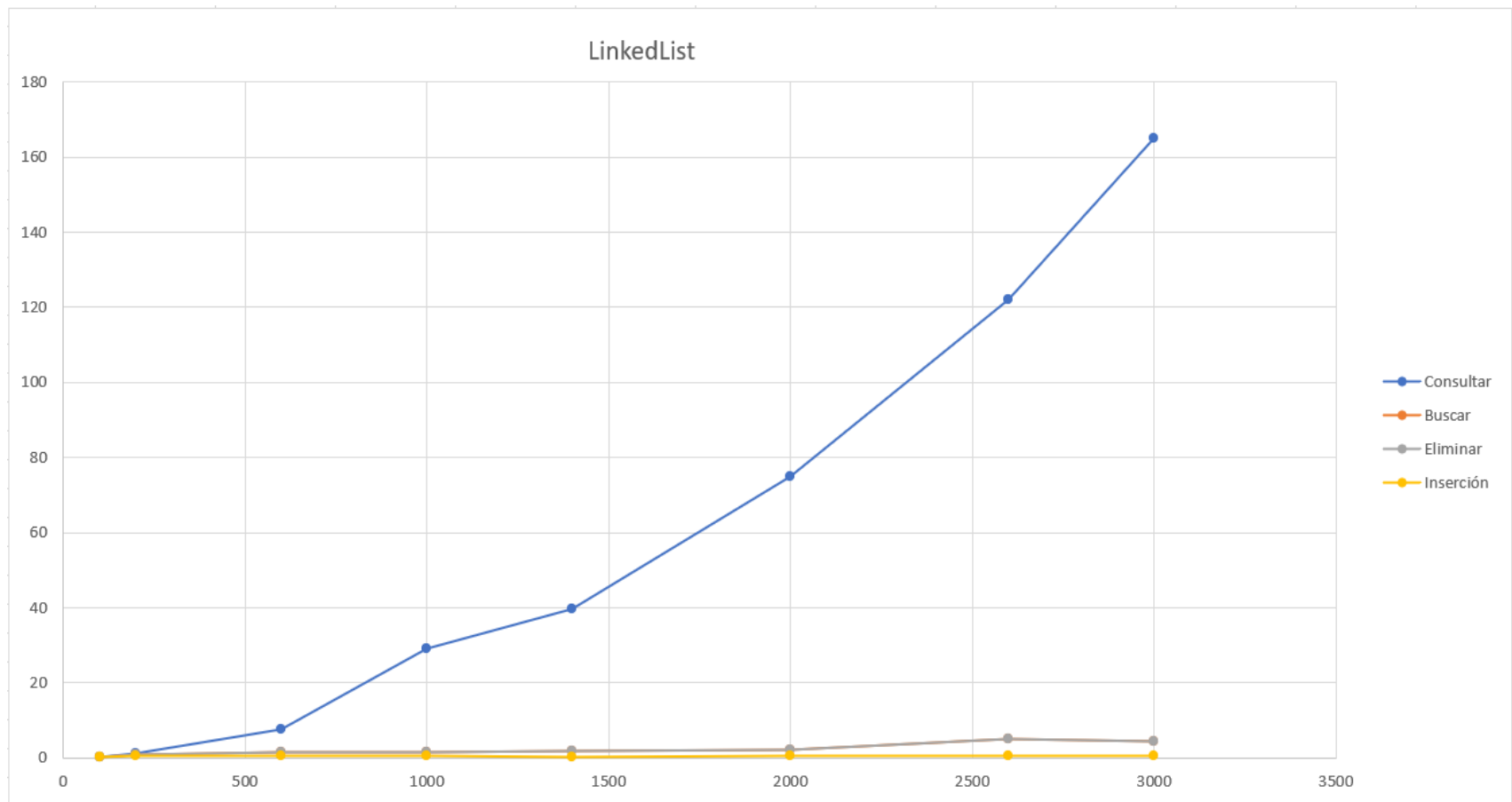
## Pruebas y análisis comparativo del uso de las estructuras de datos

ED	Array Dinámico				LinkedList			
n	1000	10000	100000	1000000	1000	10000	100000	1000000
Consulta (s)	1	1,3	1,53	26,54	57,1495	5073,7495	500739,75	50007399,7
Eliminación (s)	0,82	0,83	0,86	0,98	1,7892	15,2892	150,2892	1500,2892
Búsqueda (s)	0,8	0,84	0,85	0,89	1,7892	15,2892	150,2892	1500,2892
Inserción (s)	0,86	0,86	0,84	0,86	0,44	0,46	0,45	0,46

## Pruebas y análisis comparativo del uso de las estructuras de datos



## Pruebas y análisis comparativo del uso de las estructuras de datos







## **Pruebas y análisis comparativo del uso de las estructuras de datos**

Según las gráficas y la tabla anterior podemos concluir que para las operaciones de consulta, eliminación y búsqueda los arreglos dinámicos toman un menor tiempo para desarrollar dichas acciones pero para la operación de insertar un elemento una lista enlazada toma menos tiempo que un arreglo dinámico.