ESTRUCTURA DE DATOS

PROYECTO FINAL

INTEGRANTES:

JOSÉ ÁNGEL ROCHA GARCÍA

JESÚS DANIEL LÓPEZ HERNÁNDEZ

JOSÉ LUIS CARREÓN REYES

IVÁN ENRIQUE GONZÁLEZ RUIZ

JOSÉ MISAEL ADAME SANDOVAL

JESUS MANUEL RODRÍGUEZ RODRÍGUEZ

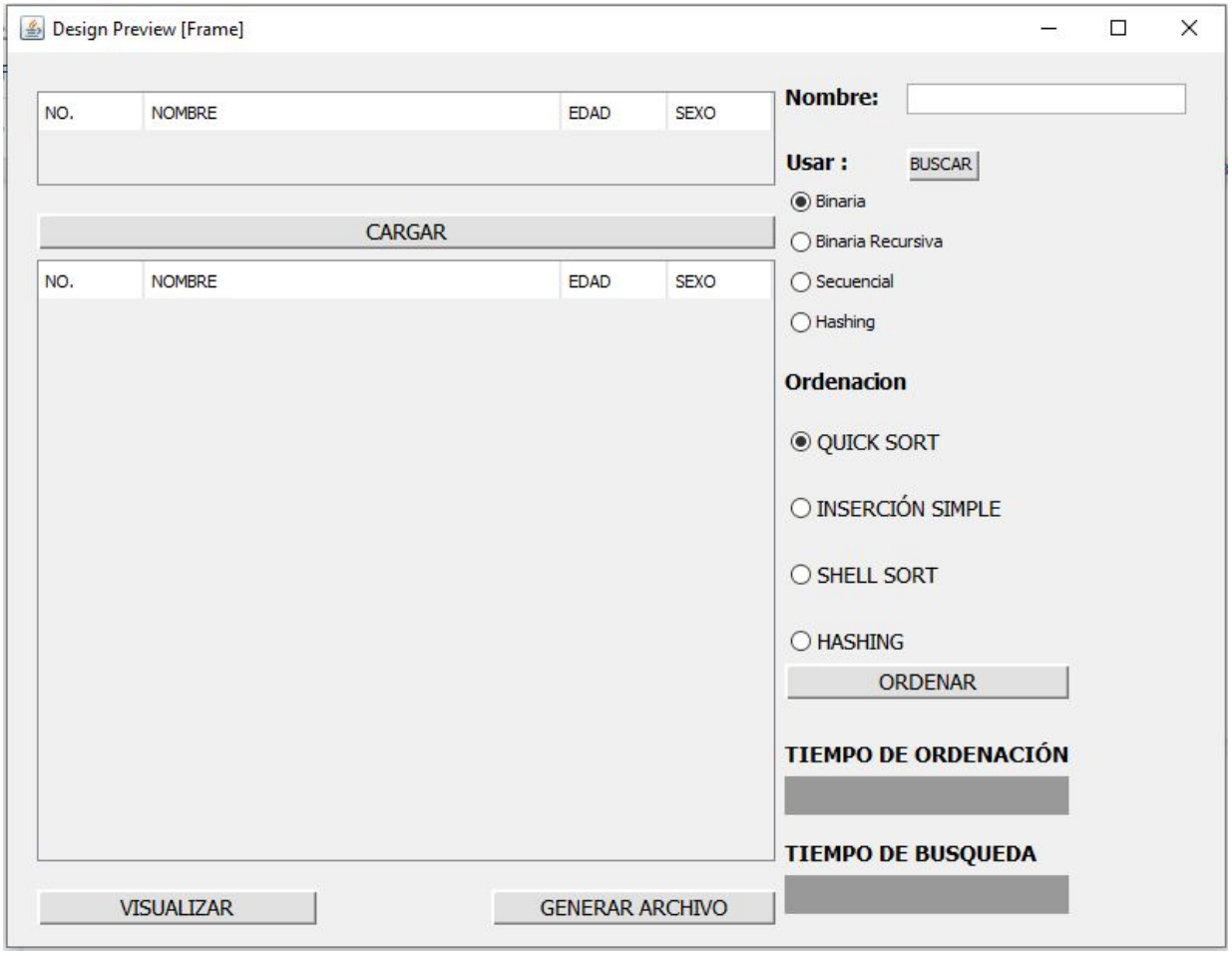
JUAN MANUEL ALFARO AGUILERA

MIGUEL IBARRA TREJO

U5-C3.- Programar la medición de tiempo para conocer la eficiencia de los algoritmos de ordenación.

U6-C3.- Programar la medición de tiempo para conocer la eficiencia de los algoritmos de búsqueda.

INTERFAZ



MÉTODO DE ORDENACIÓN	HORAS	MINUTOS	SEGUNDOS	INCREMENTOS
QuickSort	0	0	0.359	
ShellSort	0	2	50	(100000,90000,80000,70000,60000,50000,40000,30000,20000,10000,1000,100,1)
ShellSort	0	0	14.281	(1000,500,300,100,1)
ShellSort	0	0	12.3	(100000,10000,1000,100,1)
ShellSort	0	0	11.798	(50000,40000,30000,100,1)

ShellSort	0	0	10.6	(50000,40000, 30000,20000, 10000,1000,100, 1)
ShellSort	0	0	1.605	(100000,80000, 50000,30000,10 000,7000,2000,5 00,300,100,50, 20,8,3,1)
InsSimple	0	33	32.567	
Hashing	0	5	2.446	

Mejores Tiempos:
Quick Sort : 0.359 segundos
Shell Sort : 1.605 segundos, incrementos

Peores Tiempos:
Insercion Simple: 33 minutos y 32.567 segundos
Hashing : 5 minutos y 2.446 segundos

Mejores Incrementos:
(100000,80000, 50000,30000,10000,7000,2000,500,300,100,50, 20,8,3,1)
Tiempo: 1.605 segundos

(50000,40000, 30000,20000, 10000,1000,100,1)
Tiempo: 10.6 segundos

Peores Incrementos:
(100000,90000, 80000,70000,60000,50000,40000,30000,20000, 10000,1000,100,1)
Tiempo: 2 minutos y 50 segundos

(1000,500,300, 100,1)
Tiempo : 14.281 segundos

Metodo Busqueda	**Horas**	**Minutos**	**Segundos**	**NanoSeg**
Binaria	0	0	0	20000
Binaria Recursiva	0	0	0	31000
Secuencial	0	0	0	8608700
HashBusq	0	0	0.005	

Conclusion
Metodos de ordenacion

El tiempo que tarda en ser ordenado el arreglo depende mucho del algoritmo que usemos, el quick sort es el más eficaz(de ahí su nombre) mientras que los demás se acercan a el tiempo que este tarda, el shell sort es algo diferente ya que si modificamos los incrementos que recibe puede ser que se aproxime demasiado a los valores del quicksort pero también puede pasar que este tarde más que los otros. Podemos decir que todos los algoritmos tienen sus ventajas pero también sus desventajas los unos contra los otros

Metodos de busqueda

En los metodos de busqueda observamos que el método binario fue el mas eficiente, debido a su forma de dividir el arreglo, mientras que el secuencial en el peor de los casos es el más ineficiente ya que en el peor de los casos tiene que recorrer todo el arreglo,en el hashing concluimos que la eficiencia depende de que el arreglo llegue ordenado al metodo.Por lo tanto nuestra conclusion final en que dependiendo de si el arreglo esta ordenado o no, y de la posicion del elemento a buscar, siendo esta una ventaja en algunos algoritmos.