

Zundada

“ Uniendo a las personas para crear el ambiente ”

María Paula Román Arévalo

Laura Alejandra Páez Daza

Marlon David Pabon Muñoz



Descripción del Problema



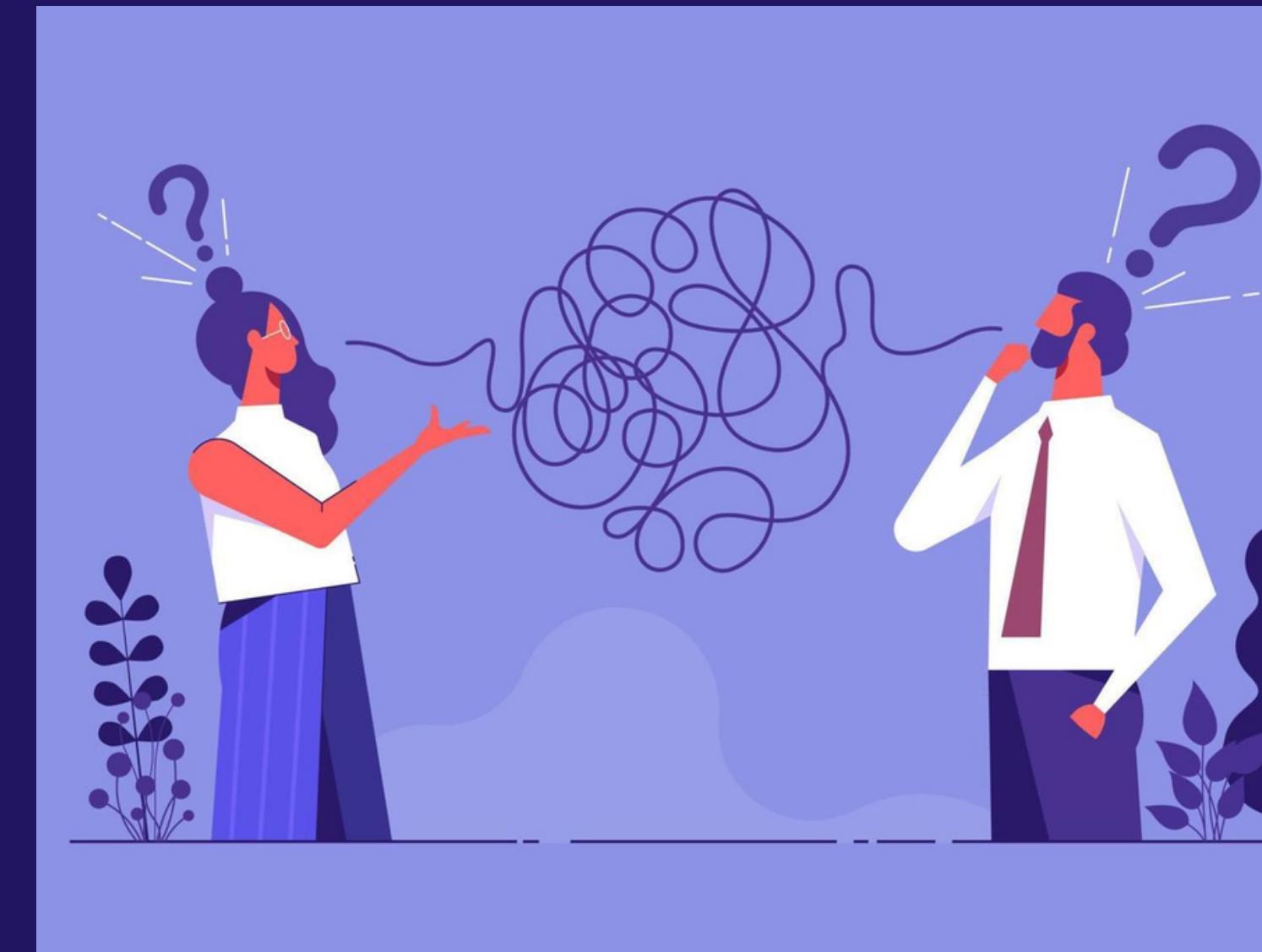
Problema

Pero estos eventos tienen una mala organización, donde destacan:

- La falta de una plataforma centralizada.
- Dificultades en la comunicación y difusión de eventos.
- Herramientas Ineficientes para el seguimiento y control de las actividades planificadas.

Contexto

El mundo universitario está lleno de actividades y eventos .



Usuarios

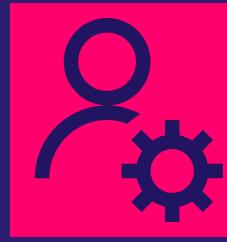
Grupo de Usuarios (Cliente, Administrador, Organizador) que utilizan la plataforma



Cliente

Función: Comprar boletas y ver eventos

Privilegios: Acceso a compra y a eventos (consultar)



Administrador

Función: Asegurar buen funcionamiento de Zundada

Privilegios: Acceso a cualquier información



Organizador

Función: Ingresar/Eliminar Eventos

Privilegios: Acceso a eventos (consultar, ingresar/eliminar)

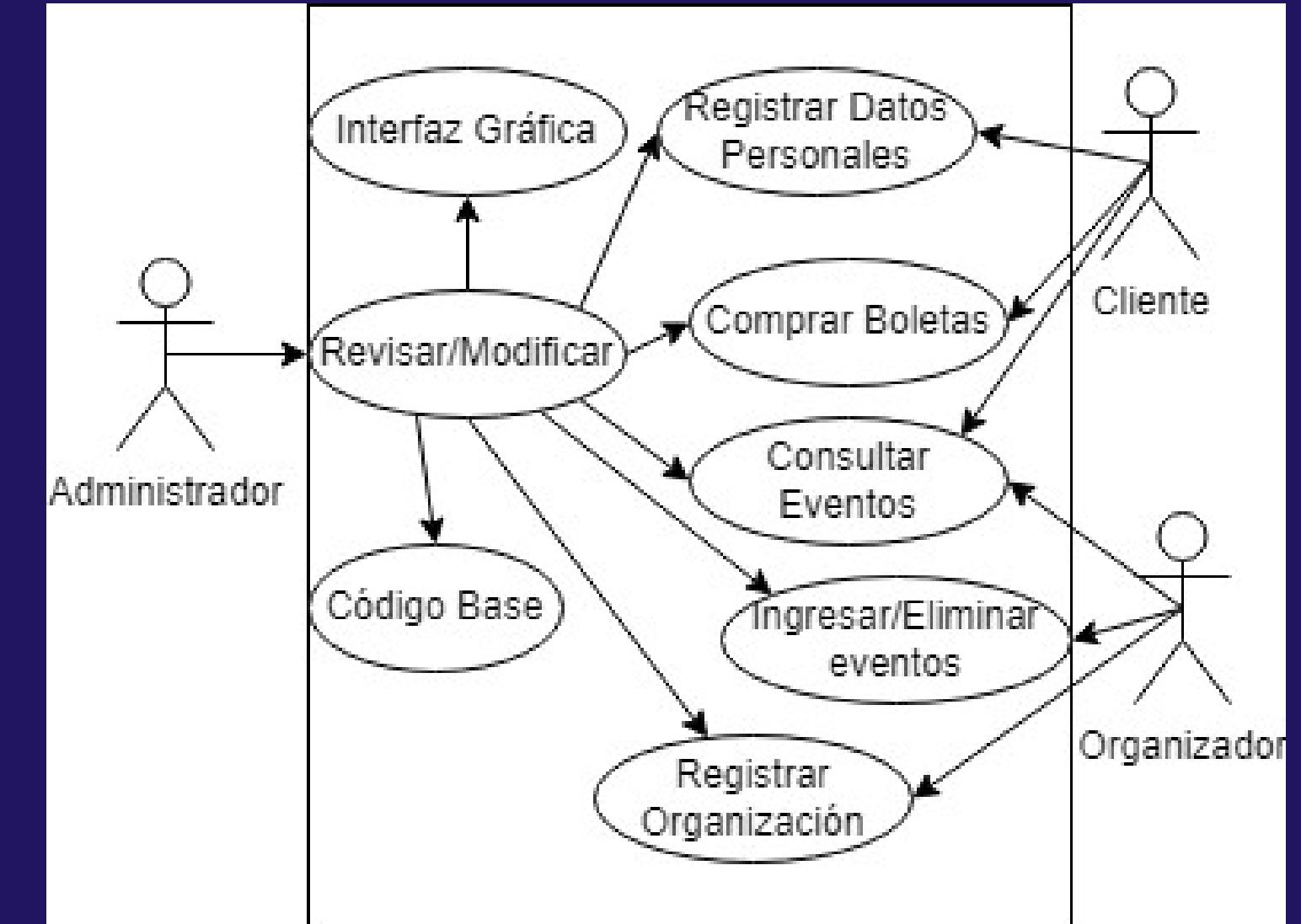


Diagrama de Casos que minimiza las funciones de cada Usuario

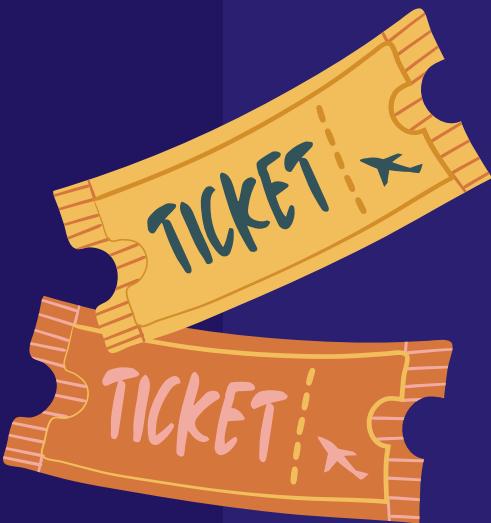


Requerimientos funcionales



Generales

- Manejo CRUD eficiente para todas las clases.
- Evitar repetición de datos.
- Objetos registrados tengan información completa.
- Página web



Eventos

- Permitir a un usuario comprar más de una boleta.
- Si se modifica o cancela el evento, enviar una notificación al cliente.
- Orden de consulta por más reciente.

Usuarios

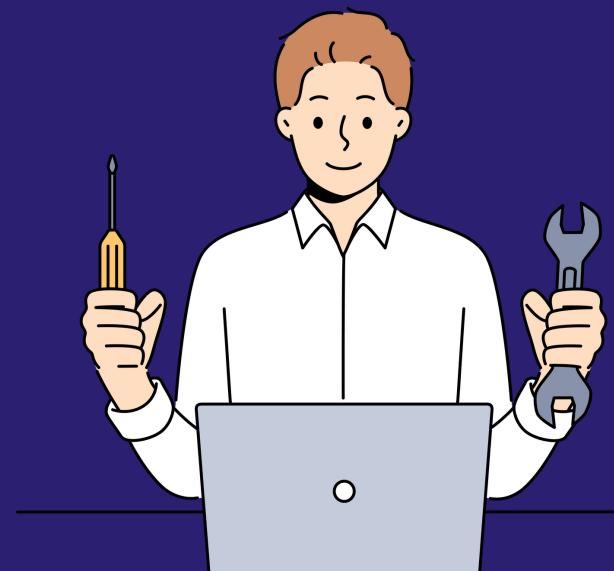
Manejo distintos roles



Cliente



Organizador



Administrador

- Restringir acceso a sus privilegios.
- Permitir varios organizadores de una misma institución.

Estructuras de datos

Usuarios



Modificar Usuario

COLA

Inserta al usuario, por medio del nombre de Usuario, y le permite modificar sus datos

- ✓ Identificar por nombreUsuario
- ✓ Verificar usuario existente
- ✓ Consulta rápida
- ✓ Modificación al ejecutar

Añadir Usuario

Array

Por medio de un array dinamico inserta almacena y consulta los usuarios

- ✓ Implementación dinámica, sin tope máximo de capacidad.
- ✓ Verificar usuario existente
- ✓ Consulta rápida



Estructuras de datos

Eventos

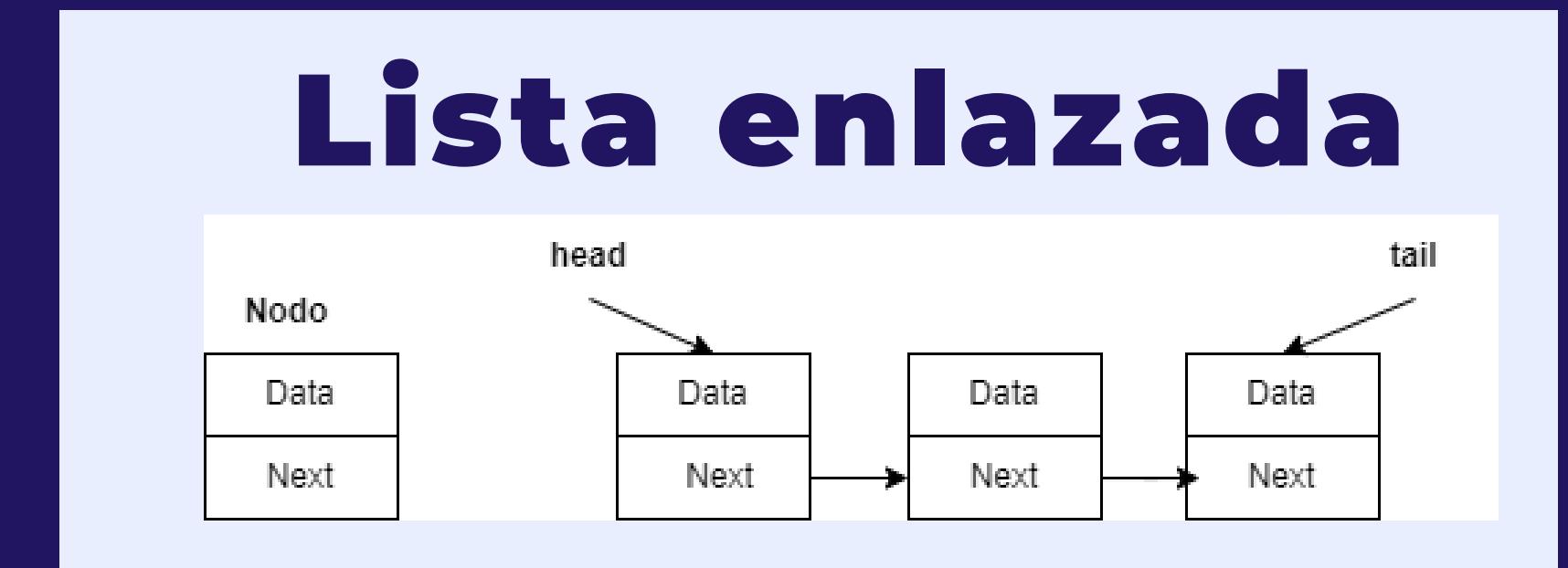
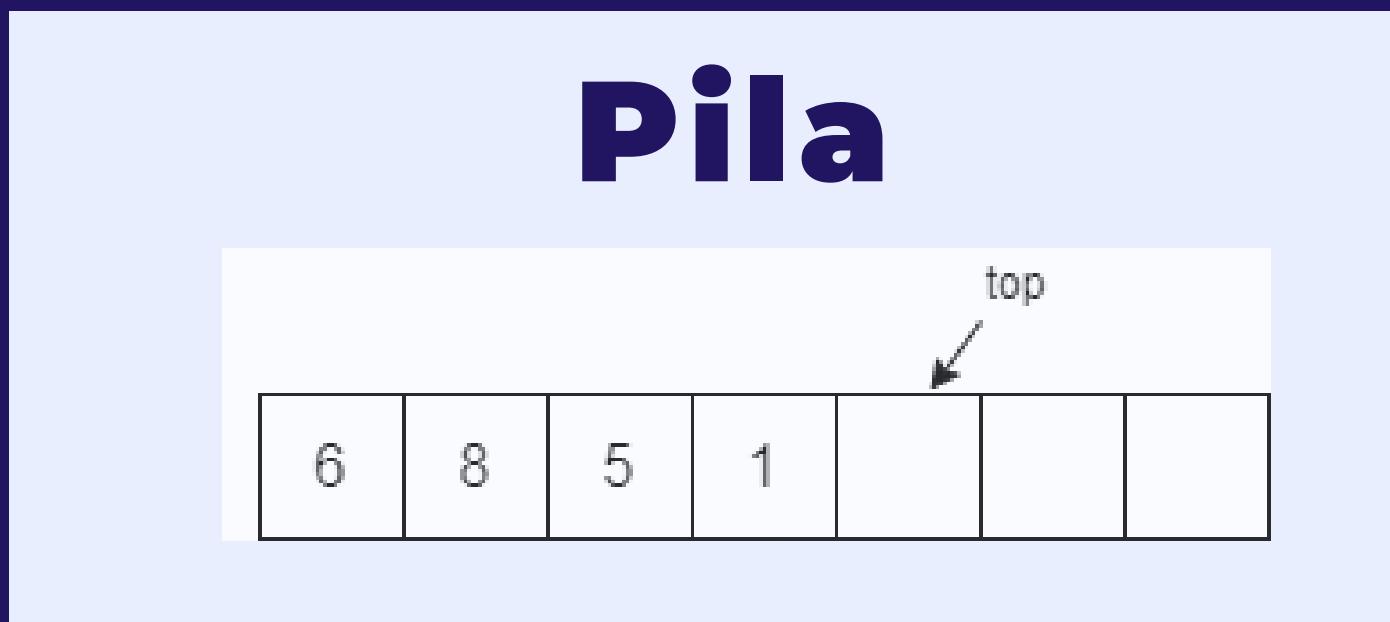


Añadir evento



Consultar eventos

eliminar eventos



Complejidad {
 $O(1)$ pop(), push(), peek()
 update()
 $O(n)$ find(), delete()

Base almacenamiento array []

Rapidez consulta por almacenamiento contiguo de memoria.

Complejidad {
 $O(1)$ pop(), push(), delete(), update()
 $O(n)$ find()

Utilización clase Nodo para almacenamiento

Rapidez inserción de datos al inicio, final o después de un nodo en específico.



Pruebas de Caso por Funcionalidad

FUNCIONALIDAD : Añadir Usuario		FUNCIONALIDAD :Añadir Usuario	
Implementación con: Array		Implementación con Lista enlazada	
Tamaño de Datos	Tiempo (ms)	Tamaño de Datos	Tiempo (ms)
1	0,5 ms	1	0,04 ms
10	0,8 ms	10	0,08 ms
100	5 ms	100	0,16 ms
1000	50 ms	1000	0,36 ms
10000	480 ms	10000	34,75 ms
100000	800 ms	100000	163,78 ms
1000000	5000 ms	1000000	3100,16 ms
10000000	60000 ms	10000000	42000 ms
100000000	240000 ms	100000000	180000 ms

FUNCIONALIDAD : Modificar Usuario	
Implementación con Colas	
Tamaño de Datos	Tiempo (ms)
10,000	4ms
100,000	40ms
1,000,000	410 ms
10,000,000	4000 ms (4s)
100,000,000	41000 ms (41s)

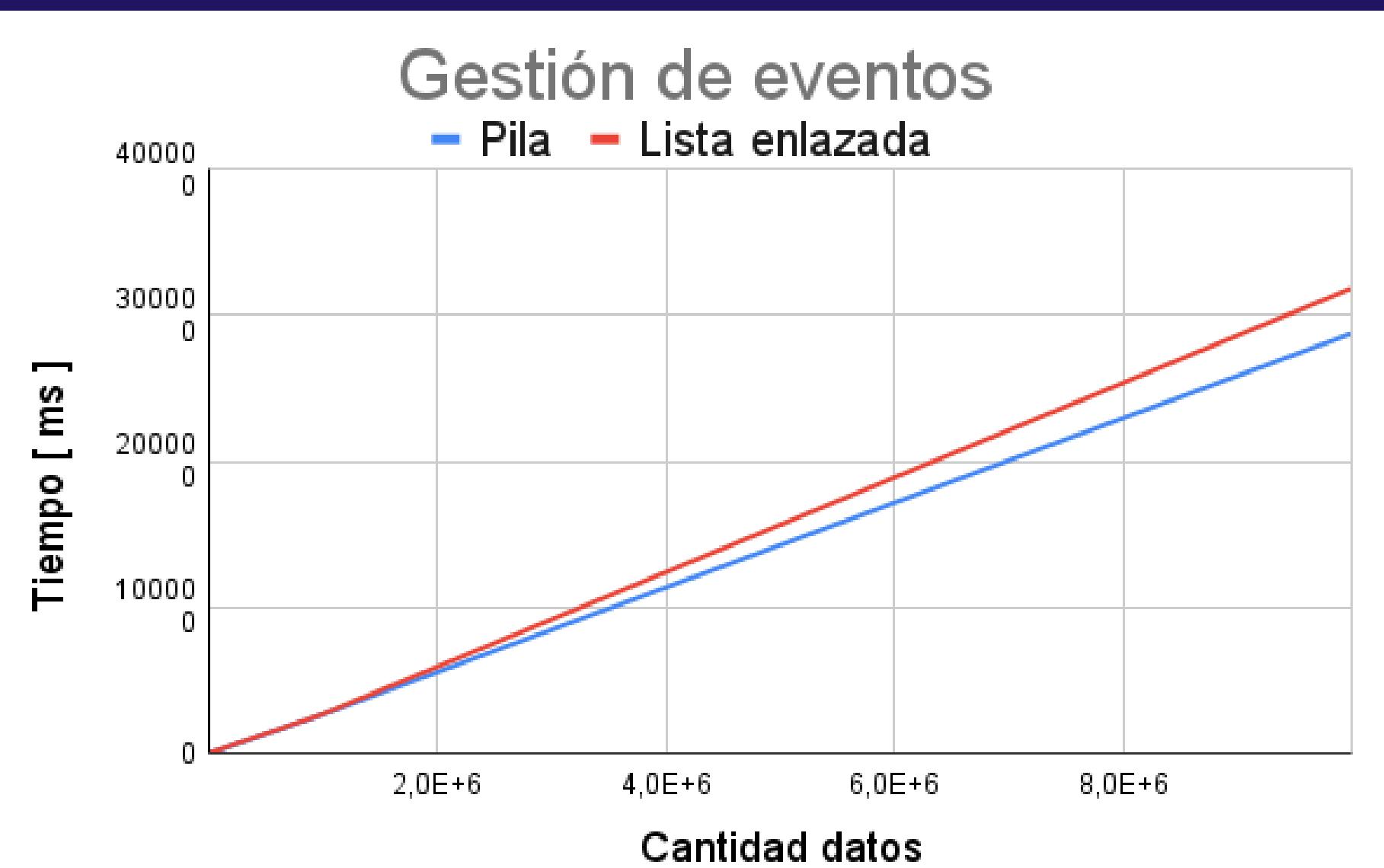
Creación de Usuario
y sus datos

Modificación
Correo Electrónico

Pruebas de Caso por Funcionalidad



Nº	Añadir Eventos		Consultar Eventos		Eliminar mitad	
	Pila	Lista	Pila	Lista	Pila	Lista
1	0,214	0,407	1,199	0,379	0,41	0,438
10	0,332	0,454	6,356	5,587	0,27	0,477
100	0,683	0,421	15,199	31,461	0,373	0,771
1000	0,913	1,714	87,535	73,737	0,414	1,153
10000	5,81	7,939	437,284	443,086	2,068	4,833
100000	19,546	11,234	2477	2784	10,098	13,697
1000000	208,334	550,469	26482	26338	13,015	19,78
10000000	3614	18776	282689	297747	860,775	1642
1000000000	Error	Error	Error	Error	Error	Error



Añadir
Evento

Consultar
Eventos

Eliminar evento en
la mitad de datos

MUCHAS GRACIAS POR LA ATENCIÓN

¿Alguna pregunta, duda, comentario o amenaza?

