OBSERVACIONES DE LA PRÁCTICA

Estudiante 1 Cod 202420470

Estudiante 2 Cod 202415759

Estudiante 3 Cod XXXX

# **Ambientes de pruebas**

|  | Máquina 1 | Máquina 2 | Máquina 3 |
| --- | --- | --- | --- |
| Procesadores | 1 | 1 | 1 |
| Memoria RAM (GB) | 32 | 24 | 16 |
| Sistema Operativo | Windows 11 | Windows 11 | Windows 10 |

*Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.*

# **Máquina 1**

## **Resultados para Queue con Array List**

| Porcentaje de la muestra | enqueue (Array List) | dequeue (Array List) | peek (Array List) |
| --- | --- | --- | --- |
| 0.50% | 0.029 | 1.050 | 0.002 |
| 5.00% | 0.324 | 11.865 | 0.003 |
| 10.00% | 0.696 | 24.788 | 0.002 |
| 20.00% | 0.645 | 49.322 | 0.002 |
| 30.00% | 0.935 | 80.245 | 0.002 |
| 50.00% | 1.154 | 136.847 | 0.002 |
| 80.00% | 2.354 | 204.356 | 0.002 |
| 100.00% | 3.456 | 296.355 | 0.005 |

## **Resultados para Stack con Array List**

| Porcentaje de la muestra | push (Array List) | pop (Array List) | top(Linked List) |
| --- | --- | --- | --- |
| 0.50% | 0.34 | 0.068 | 0.002 |
| 5.00% | 0.455 | 0.578 | 0.002 |
| 10.00% | 0.970 | 1.124 | 0.003 |
| 20.00% | 1.076 | 1.679 | 0.003 |
| 30.00% | 1.579 | 1.578 | 0.003 |
| 50.00% | 1.079 | 1.106 | 0.002 |
| 80.00% | 2.778 | 1.848 | 0.002 |
| 100.00% | 5.785 | 2.157 | 0.002 |

## **Resultados para Queue con Linked List**

| Porcentaje de la muestra | enqueue (Linked List) | dequeue (Linked List) | peek Linked List) |
| --- | --- | --- | --- |
| 0.50% | 0.035 | 0.046 | 0.002 |
| 5.00% | 0.339 | 0.544 | 0.003 |
| 10.00% | 0.657 | 1.192 | 0.003 |
| 20.00% | 0.600 | 6.967 | 0.003 |
| 30.00% | 0.958 | 17.07 | 0.002 |
| 50.00% | 1.451 | 32.740 | 0.002 |
| 80.00% | 2.354 | 111.096 | 0.003 |
| 100.00% | 3.235 | 101.813 | 0.006 |

## **Resultados para Stack con Linked List**

| Porcentaje de la muestra | push (Linked List) | pop (Linked List) | top(Linked List) |
| --- | --- | --- | --- |
| 0.50% | 0.034 | 0.046 | 0.002 |
| 5.00% | 0.447 | 0.518 | 0.003 |
| 10.00% | 0.960 | 1.028 | 0.002 |
| 20.00% | 1.080 | 1.032 | 0.003 |
| 30.00% | 1.591 | 1.457 | 0.002 |
| 50.00% | 1.070 | 1.011 | 0.001 |
| 80.00% | 2.745 | 1.951 | 0.001 |
| 100.00% | 25.565 | 0.024 | 0.003 |

# **Máquina 2**

## **Resultados para Queue con Array List**

| Porcentaje de la muestra | enqueue (Array List) | dequeue (Array List) | peek (Array List) |
| --- | --- | --- | --- |
| 0.50% | 0.011 | 0.007 | 0.002 |
| 5.00% | 0.072 | 0.062 | 0.024 |
| 10.00% | 0.020 | 0.014 | 0.003 |
| 20.00% | 0.055 | 0.044 | 0.002 |
| 30.00% | 0.041 | 0.049 | 0.002 |
| 50.00% | 0.064 | 0.063 | 0.002 |
| 80.00% | 0.09 | 0.097 | 0.002 |
| 100.00% | 0.152 | 0.155 | 0.002 |

## **Resultados para Stack con Array List**

| Porcentaje de la muestra | push (Array List) | pop (Array List) | top(Array List) |
| --- | --- | --- | --- |
| 0.50% | 0.011 | 0.01 | 0.004 |
| 5.00% | 0.037 | 0.04 | 0.007 |
| 10.00% | 0.015 | 0.011 | 0.003 |
| 20.00% | 0.04 | 0.035 | 0.003 |
| 30.00% | 0.042 | 0.025 | 0.003 |
| 50.00% | 0.065 | 0.063 | 0.003 |
| 80.00% | 0.059 | 0.055 | 0.003 |
| 100.00% | 0.08 | 0.08 | 0.003 |

## **Resultados para Queue con Linked List**

| Porcentaje de la muestra | enqueue (Linked List) | dequeue (Linked List) | peek Linked List) |
| --- | --- | --- | --- |
| 0.50% | 0.100 | 0.029 | 0.001 |
| 5.00% | 6.532 | 0.29 | 0.001 |
| 10.00% | 10.821 | 0.53 | 0.001 |
| 20.00% | 20.345 | 0.82 | 0.001 |
| 30.00% | 60.678 | 1.24 | 0.001 |
| 50.00% | 100.342 | 1.89 | 0.001 |
| 80.00% | 130.403 | 2.13 | 0001 |
| 100.00% | 201.32 | 3.18 | 0.002 |

## **Resultados para Stack con Linked List**

| Porcentaje de la muestra | push (Linked List) | pop (Linked List) | top(Linked List) |
| --- | --- | --- | --- |
| 0.50% | 0.053 | 0.027 | 0.001 |
| 5.00% | 2.31 | 0.28 | 0.001 |
| 10.00% | 10.24 | 0.456 | 0.001 |
| 20.00% | 40.56 | 0.853 | 0.001 |
| 30.00% | 50.40 | 1.023 | 0.001 |
| 50.00% | 92.34 | 1.843 | 0.001 |
| 80.00% | 113.02 | 3.453 | 0.001 |
| 100.00% | 160.32 | 4.232 | 0.001 |

# **Máquina 3**

## **Resultados para Queue con Array List**

| Porcentaje de la muestra | enqueue (Array List) | dequeue (Array List) | peek (Array List) |
| --- | --- | --- | --- |
| 0.50% | **0.042** | **0.04** | **0.002** |
| 5.00% | **0.197** | **0.357** | **0.002** |
| 10.00% | **0.396** | **0.785** | **0.002** |
| 20.00% | **0.789** | **1.588** | **0.002** |
| 30.00% | **1.324** | **2.771** | **0.002** |
| 50.00% | **2.006** | **4.686** | **0.002** |
| 80.00% | **3.335** | **9.01** | **0.002** |
| 100.00% | **4.349** | **12.654** | **0.002** |

## **Resultados para Stack con Array List**

| Porcentaje de la muestra | push (Array List) | pop (Array List) | top(Array List) |
| --- | --- | --- | --- |
| 0.50% | **0.042** | **0.034** | **0.002** |
| 5.00% | **0.327** | **0.322** | **0.002** |
| 10.00% | **0.675** | **0.636** | **0.001** |
| 20.00% | **1.304** | **1.492** | **0.001** |
| 30.00% | **2.08** | **2.357** | **0.002** |
| 50.00% | **3.443** | **3.227** | **0.001** |
| 80.00% | **34.404** | **5.406** | **0.001** |
| 100.00% | **7.038** | **6.43** | **0.001** |

(Nota, el 34.404 NO es un error de tipeo, es el valor que dio la máquina)

## **Resultados para Queue con Linked List**

| Porcentaje de la muestra | enqueue (Linked List) | dequeue (Linked List) | peek Linked List) |
| --- | --- | --- | --- |
| 0.50% | **0.101** | **0.037** | **0.002** |
| 5.00% | **7.616** | **0.377** | **0.002** |
| 10.00% | **31.783** | **0.775** | **0.002** |
| 20.00% | **130.547** | **1.565** | **0.002** |
| 30.00% | **301.978** | **2.566** | **0.002** |
| 50.00% | **852.327** | **5.052** | **0.002** |
| 80.00% | **2200.431** | **8.953** | **0.002** |
| 100.00% | **3404.065** | **12.373** | **0.004** |

## **Resultados para Stack con Linked List**

| Porcentaje de la muestra | push (Linked List) | pop (Linked List) | top(Linked List) |
| --- | --- | --- | --- |
| 0.50% | **0.103** | **0.031** | **0.002** |
| 5.00% | **7.371** | **0.318** | **0.001** |
| 10.00% | **31.879** | **0.732** | **0.002** |
| 20.00% | **132.598** | **1.259** | **0.002** |
| 30.00% | **313.015** | **2.404** | **0.002** |
| 50.00% | **856.535** | **3.637** | **0.001** |
| 80.00% | **2212.49** | **5.877** | **0.002** |
| 100.00% | **3423.624** | **7.46** | **0.003** |

# **Preguntas de análisis**

1. ¿Se observan diferencias significativas entre las implementaciones con ArrayList y LinkedList para las funciones de Queue y Stack? ¿Cuál es más eficiente en cada operación? ¿Por qué una implementación es más rápida en ciertos casos?

Si, aunque en el stack se comportan de manera similar, aunque el linkedlist mantiene ventaja debido a que su memoria no contigua le beneficia a la hora de agregar elementos, a diferencia del arraylist con su estricta estructura.

En el dequeue se observan diferencias por ejemplo, para el stack el single linked es mucho más eficiente pues es solo O(1), esto ocurre porque en arraylist, la función Pop tiene una complejidad de O(N).

1. ¿Cuándo es preferible usar ArrayList o LinkedList? Si insertamos y eliminamos con frecuencia, ¿qué estructura conviene más? Si accedemos aleatoriamente a elementos, ¿cuál es más eficiente?

Para el primer caso es más eficiente linkedlist pues permite mayor facilidad a la hora de agregar o eliminar elementos en la lista. Para el otro caso es mejor arraylist pues su acceso a elementos es inmediato, lo cúal facilita el trabajo en un ambiente de acceso aleatorio.

1. Durante la ejecución de las pruebas ¿Se presentan anomalías en los tiempos de ejecución que no se explican con la teoría?

Si en algunos casos se demoraba más sin importar que el porcentaje sea menor, teorizamos que es a algún proceso de los equipos el cúal no conocemos.

1. Complete la siguiente tabla de acuerdo con qué operación es más eficiente en cada implementación (marque con una x la que es más eficiente). Adicionalmente, explique si este comportamiento es acorde con lo enunciado en la teoría. Justifique las respuestas.

|  |  | Array List | Linked List | Justificación |
| --- | --- | --- | --- | --- |
| QUEUE | **Enqueue()** | x | x | Ambas son O(1) |
| **Dequeue()** |  | x | arraylist es O(N) y singlelinked es O(1) |
| **Peek()** | x | x | Ambas son O(1) |
| STACK | **Push()** | x | x | Ambas son O(1) |
| **Pop()** | x | x | Ambas son O(1) |
| **Top()** | x | x | Ambas son O(1) |

Esto es de acuerdo a la teoría vista en clase, y se debe a las diferencias en cuanto al funcionamiento de estos arreglos.