**Método de la ingeniería**

***Fase 1: Identificación del problema.***

* **Problema**

Actualmente las bosa de valores de Colombia (BVC) no permite tranzar con acciones internacionales ni trabajar con el mercado de divisas o derivados. Debido a esto la BVC quiere aprovechar estas coyunturas y quiere consolidar una aplicación, que permita manejar los datos de algunos mercados de divisas y de acciones internacionales.

La empresa desea que la aplicación tenga los siguientes requerimientos funcionales y no funcionales.

* **Requerimientos Funcionales**

|  |  |
| --- | --- |
| **Nombre** | R1 |
| **Resumen** | Consultar el precio más alto de una acción o un mercado de divisas en un rango de tiempo. |
| **Entrada** | Rango de tiempo |
| **Salida** | Precio más alto de la acción o de la divisa |

|  |  |
| --- | --- |
| **Nombre** | R2 |
| **Resumen** | Consultar el precio más bajo de una acción o un mercado de divisas en un rango de tiempo. |
| **Entrada** | Rango de tiempo |
| **Salida** | Precio más bajo de la acción o de la divisa |

|  |  |
| --- | --- |
| **Nombre** | R3 |
| **Resumen** | Consultar el periodo de tiempo donde una acción / mercado de divisas tuvo su mayor crecimiento. |
| **Entrada** | Acción / Mercado de divisas |
| **Salida** | Periodo de tiempo con mayor crecimiento |

|  |  |
| --- | --- |
| **Nombre** | R4 |
| **Resumen** | Permitir ver cuáles acciones / Mercado de divisas superan un valor en un rango de tiempo. |
| **Entrada** | Rango de tiempo  Valor a superar |
| **Salida** | Acciones o Mercado de divisas que superan ese valor en un determinado periodo de tiempo |

|  |  |
| --- | --- |
| **Nombre** | R5 |
| **Resumen** | Consultar cuales son las 3 acciones / Mercados que presentaron mayor crecimiento en un rango de tiempo. |
| **Entrada** | Rango de tiempo |
| **Salida** | Acciones o Mercado de divisas con mayor crecimiento en un rango de tiempo |

|  |  |
| --- | --- |
| **Nombre** | R6 |
| **Resumen** | Ingresar datos por medio de archivos de texto |
| **Entrada** | Archivo de texto |
| **Salida** | Datos Ingresados |

|  |  |
| --- | --- |
| **Nombre** | R7 |
| **Resumen** | Ingresar datos de manera individual |
| **Entrada** | Acción / Divisas / Bitcoin |
| **Salida** | Dato Ingresado |

* **Requerimientos no funcionales**

|  |  |
| --- | --- |
| **Nombre** | **RFN1** |
| **Resumen** | Para el mercado de divisas su complejidad de las operaciones básicas debe ser O(Log n) |

|  |  |
| --- | --- |
| **Nombre** | **RFN2** |
| **Resumen** | Para el mercado de acciones su complejidad de las operaciones básicas debe ser O(Log n) |

|  |  |
| --- | --- |
| **Nombre** | **RFN3** |
| **Resumen** | Para el mercado de BitCoins su complejidad de las operaciones básicas debe ser O(n) |

|  |  |
| --- | --- |
| **Nombre** | **RFN4** |
| **Resumen** | Mostrar una gráfica del estado de los precios de una acción con un color |

|  |  |
| --- | --- |
| **Nombre** | **RFN5** |
| **Resumen** | Mostrar una gráfica del estado de los precios de un mercado de divisas con un color |

***Fase 2: Recopilación de la información.***

***Fase 3: Búsqueda de soluciones creativas.***

Para la solución de este problema necesitamos enfocarnos en las estructuras de datos que se adecuan más para almacenar las acciones, divisas y bitcoins, esto nos dará que el programa sea eficiente y preciso a la hora de dar la solución.

* **Alternativa 1**

Esta alternativa se basa en almacenar las acciones en un árbol AVL, las divisas en un árbol Rojo-Negro y por último los bitcoins en una lista enlazada.

Divisa 1

Divisa 6

Divisa 5

Divisa 4

Divisa 3

Divisa 2

F.B = 0

F.B = 1

F.B = 0

F.B = 0

F.B = 0

F.B = 0

Acción 1

Acción 1

Acción 1

Acción 1

Acción 1

Acción 1



null

* **Alternativa 2**

En esta alternativa utilizaremos listas enlazadas para conectar las acciones, divisas y bitcoins.



null

null

Acción 3

Acción 2

Acción 1

null

Divisa 3

Divisa 2

Divisa 1

* **Alternativa 3**

En esta alternativa se optará por almacenar las acciones en una pila, las divisas en una cola y los bitcoins en TablaHash.

.

Acción 1

Acción 2

Acción 3

Acción 4

B2

B3

B1

B4

Divisa 1

Divisa 2

Divisa 3

Divisa 4

Divisa 5

* **Alternativa 4**

Esta alternativa se basa en poner las acciones, divisas y bitcoins en arreglos para así poder tener acceso a ellos de manera lineal.

Divisa 1

Divisa 2

Divisa 3

Divisa 4

Divisa 5

Acción 1

Acción 2

Acción 3

Acción 4

Acción 5

Bitcoin 1

Bitcoin 2

Bitcoin 3

Bitcoin 4

Bitcoin 5

* **Alternativa 5**

Esta alternativa se basa en poner las acciones, divisas y bitcoins en tres tablas hash y así poder acceder cualquier acción, divisa o bitcoin en tiempo constante.

B3

B1

B2

B4

D3

D1

D2

D4

A3

A1

A2

A4

* **Alternativa 6**

En esta alternativa pondremos todas las acciones, divisas y bitcoins en pilas así podremos obtener los últimos elementos que se agregaron en tiempo constante.

Acción 1

Acción 2

Acción 3

Acción 4

Divisa 1

Divisa 2

Divisa 3

Divisa 4

Bitcoin 1

Bitcoin 2

Bitcoin 3

Bitcoin 4

* **Alternativa 7**

En esta alternativa pondremos todas las acciones, divisas y bitcoins en colas así podremos obtener los primeros y últimos elementos que se agregaron en tiempo constante.

Divisa 1

Divisa 2

Divisa 3

Divisa 4

Divisa 5

Acción 1

Acción 2

Acción 3

Acción 4

Acción 5

Bitcoin1

Bitcoin 2

2

Bitcoin 3

Bitcoin 4

Bitcoin 5

***Fase 4: Transición de la formulación de ideas a los diseños preliminares.***

* **Descarte de ideas no factibles**

Se descartaron las siguientes opciones de la búsqueda de soluciones creativas debido a:

|  |  |
| --- | --- |
| **Alternativa 3** | Se descartó esta alternativa debido a que almacenar las acciones en una pila sólo nos permitiría obtener la última acción y para obtener las otras tendríamos que eliminar todas hasta que obtengamos la que queremos.  Almacenar las divisas en una cola no tendría sentido porque sólo podemos conocer la primera y la última divisa y para conocer las demás tendríamos que eliminar divisas.  Almacenar los bitcoins en una TablaHash se podría hacer, pero no va con el contexto del programa que el cliente pidió. |
| **Alternativa 4** | Esta alternativa se descarta porque almacenar todas las divisas, acciones y bitcoins en un arreglo es una solución que para la gran cantidad de datos que vamos a almacenar resultaría muy ineficiente. |
| **Alternativa 5** | Esta alternativa se descarta ya que almacenar todos las divisas, acciones y bitcoins en HashTables es eficiente con respecto al tiempo, no tiene nada que ver con el contexto del problema. |
| **Alternativa 6** | Esta alternativa se descarta debido a que almacenar todos los datos en pilas, no resultaría nada eficiente en complejidad espacial y temporal. |
| **Alternativa 7** | Esta alternativa se descarta debido a que almacenar todos los datos en colas, no resultaría nada eficiente en complejidad espacial y temporal, además, no podríamos consultar los datos sin tender que eliminar otros asiendo que perdamos información. |

***Fase 5: Evaluación y selección de la mejor solución.***

Actualmente tenemos dos alternativas de solución, las cuales resuelven nuestro problema, como sólo necesitamos una alternativa, definiremos una serie de criterios que nos ayudaran a escoger la mejor solución para nuestro problema.

* **Criterios**
* **Criterio A: Representación fiel al contexto del problema.**

Este criterio se basa en que tan representativa es la alternativa al contexto del problema.

* Fiel: 3 puntos
* Regular: 2 puntos
* Mala: 1 punto
* **Criterio B: Complejidad Espacial en Acciones y Divisas**

Este criterio se basa en que tan eficiente en espacio de memoria es la alternativa para las acciones y divisas para la solución del problema.

* Constante: 6 puntos
* Logarítmica: 5 puntos
* Lineal: 4 puntos
* Polinomial: 3 puntos
* Exponencial: 2 puntos
* Factorial: 1 punto
* **Criterio C: Complejidad Temporal en Acciones y Divisas**

Este criterio se basa en que tan eficiente es en tiempo la alternativa para las acciones y divisas para la solución del problema.

* Constante: 6 puntos
* Logarítmica: 5 puntos
* Lineal: 4 puntos
* Polinomial: 3 puntos
* Exponencial: 2 puntos
* Factorial: 1 punto
* **Criterio D: Complejidad Temporal en Bitcoins**

Este criterio se basa en que tan eficiente es en tiempo la alternativa para los bitcoins.

* Constante: 6 puntos
* Logarítmica: 5 puntos
* Lineal: 4 puntos
* Polinomial: 3 puntos
* Exponencial: 2 puntos
* Factorial: 1 punto
* **Criterio E: Complejidad Espacial en Bitcoins**

Este criterio se basa en que tan eficiente en espacio de memoria es la alternativa para los bitcoins.

* Constante: 6 puntos
* Logarítmica: 5 puntos
* Lineal: 4 puntos
* Polinomial: 3 puntos
* Exponencial: 2 puntos
* Factorial: 1 punto
* **Evaluación:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Criterio A** | **Criterio B** | **Criterio C** | **Criterio D** | **Criterio E** | **Total** |
| **Alternativa 1** | 3 | 4 | 5 | 4 | 4 | 20 |
| **Alternativa 2** | 2 | 4 | 4 | 4 | 4 | 18 |

Con base en los resultados obtenidos la alternativa 2 va a ser descartada y por tanto la alternativa 1 será la utilizada para dar solución al problema.