1. **Identificación del problema**

**Contexto Problemático**

La Bolsa de Valores de Colombia (BVC) es una bolsa multi-producto y multi-mercado que administra los sistemas de negociación y registro de los mercados de acciones, renta fija, derivados, divisas, OTC y servicios de emisores en Colombia. Actualmente la BVC no permite transar con acciones internacionales ni trabajar con el mercado de divisas o de derivados, sin embargo la BVC con la ayuda del gobierno quiere consolidar una aplicación que contengan los datos de algunos mercados de divisas y de acciones internacionales para esto una compañía desarrolladora de software de mediana escala, pero de mucho futuro, ha solicitado la implementación de la aplicación de la BVC.

**Identificación del problema**

La implementación de la aplicación de la BVC necesita, para el manejo de información de gran tamaño, que permita ingresar datos, eliminar o modificar datos, realizar consultas de mercados de capitales y mercados de divisas. Entonces, de acuerdo a lo anterior, las especificaciones que debe cumplir el programa a desarrollar son:

1. Consultar el precio más alto de una acción o un mercado de divisas en un rango de tiempo. Es decir, el precio más alto de la misma, dada una fecha inicial y una fecha final.
2. Consultar el precio más bajo de una acción o un mercado de divisas en un rango de tiempo. Es decir, el precio más alto de la misma dada una fecha inicial y una fecha final.
3. Consultar el periodo de tiempo donde una acción / mercado de divisas tuvo su mayor crecimiento.
4. Mostrar una gráfica del estado de los precios de una acción / mercado de divisas. En la gráfica debe ser posible agregar hasta un máximo de 3 acciones / mercados de divisas en donde cada indicador de gráfica deberá tener un color diferente.
5. Mostrar cuáles acciones / Mercado de divisas superan un valor en un rango de tiempo.
6. Mostrar cuales son las 3 acciones / Mercados que presentaron mayor crecimiento en un rango de tiempo.

**2. Recopilación de la información necesaria**

**Mercado de divisas:** El mercado de divisas, también conocido como Forex (abreviatura del término inglés Foreign Exchange), FX o Currency Market, es un mercado mundial y descentralizado en el que se negocian divisas.

Este mercado nació con el objetivo de facilitar el [flujo monetario](https://es.wikipedia.org/wiki/Flujo_monetario) que se deriva del comercio internacional. Es, por gran margen, el mercado financiero más grande del mundo, llegando a mover un volumen diario de transacciones de alrededor de cinco billones de dólares estadounidenses (USD), más que todos los demás mercados bursátiles del planeta combinados. Ha crecido tanto que, en la actualidad, el total de operaciones en moneda extranjera que se debe a operaciones internacionales de bienes y servicios representan un porcentaje casi residual, debiéndose la mayoría de las mismas a compraventa de activos financieros.​ En consecuencia este mercado es bastante independiente de las operaciones comerciales reales y las variaciones entre el precio de dos monedas no puede explicarse de forma exclusiva por las variaciones de los flujos comerciales.

Según el Banco de Pagos Internacionales, los resultados mundiales preliminares de la Encuesta de Operaciones de Divisas y de Derivados del Mercado Bancario Trienal de 2016 muestran que el comercio en los mercados de divisas promedió $ 5,09 billones por día en abril de 2016. Esto es inferior a $ 5.4 billones en en abril de 2013, pero en comparación con los 4,0 billones de dólares en abril de 2010. Medido por valor, los de cambio de divisas se negociaron más que cualquier otro instrumento en abril de 2016, a 2,4 billones de dólares por día, seguido del comercio al contado a 1,7 billones de dólares.

**XAUUSD:** Es el código para el Oro en la tabla periódica de elementos, y el precio arriba indicado corresponde a la cotización del oro en dólares norteamericanos, que es el criterio común para medir su valor alrededor del mundo. (Tradingview, 2019)

**ERUSUD:** Se define (EUR/USD) como la abreviatura para el par de divisas o cruce del euro y el dólar estadounidense, las monedas de la Zona Euro (EUR) y Estados Unidos (USD). Este par de divisas muestra cual es la cantidad de dólares estadounidenses (la divisa de cotización) que se requiere para comprar un euro (la divisa base). El valor o cotización del par EUR/USD es mostrado en términos de 1 euro por una cantidad determinada de dólares estadounidenses, es decir el valor de 1 EUR en USD. En otras palabras, el valor del EUR/USD indica cual es la cotización del euro en términos del dólar estadounidense. Por ejemplo, si el par se negocia a 1,1500 esto significa que se necesita 1,1500 dólares estadounidenses para comprar 1 euro. Negociar con el par de divisas EUR/USD también se conoce como operar con el “Euro”. (técnicasdetrading, 2019)

**GBPCAD:** GBP/CAD es la abreviatura del par de la libra esterlina y el dólar canadiense. Muestra cuánto vale el GBP (moneda base) medido frente al CAD (moneda contraria). Por ejemplo, GBP/CAD = 1.6685 indica que una libra esterlina puede comprar 1.6685 dólares canadienses. (fxstreet, 2019).

**USDJPY:** el par de divisas USD/JPY (dólar estadounidense/yen japonés) es uno de los más negociados en los mercados internacionales de divisas. Los inversores y especuladores interesados en operar con este par deben analizar con cuidado la economía japonesa y la evolución de las cotizaciones del USD/JPY con el fin de determinar si este es un instrumento financiero que se adapta a sus estrategias y necesidades de inversión. Como veremos a continuación, el USD/JPY tiene características que lo convierten en un mercado interesante para muchos traders, incluyendo especuladores, sin embargo en ocasiones puede resultar demasiado riesgoso para los traders que tienen una alta aversión al riesgo ya que presenta períodos de elevada volatilidad, aunque no tanto como otros pares relacionados con el yen como el EUR/JPY o el GBP/JPY. (tecnicasdetrading, 2019).

**Mercado de acciones:** El Capital Social de una empresa representa el capital originalmente pagado o invertido en el negocio por sus fundadores. Sirve como garantía para los acreedores de una empresa, ya que no pueden ser retirados en detrimento de los acreedores. El capital social de una empresa se divide en acciones, el total de los que deben declararse en el momento de la creación de empresas. Según la cantidad total de dinero invertido en el negocio, cada acción tiene un cierto valor nominal, comúnmente conocido como el valor nominal de la acción. Las acciones representan una fracción de la propiedad en una empresa. (Perez, 2018)

**US30:** El US 30 es un instrumento financiero popular que se basa en el desempeño del índice Dow Jones Industrial Average Future. Puede cambiarlo en cualquier dirección, hacia arriba o hacia abajo, utilizando apalancamiento de hasta 200:1. iFOREX ofrece la oportunidad de invertir en el US 30, que se basa en el rendimiento del índice futuro del Dow Jones, en forma de CFD. El Dow Jones Índice Industrial es uno de los índices bursátiles estadounidenses más famosos y seguidos de cerca. Fue fundado en 1896 por Charles Dow, editor del Wall Street Journal y co-fundador de US 30 & Company. Contiene 30 de las empresas más grandes cotizadas en bolsa con sede en los Estados Unidos de América. El índice está basado en un promedio ponderado de precios, y para compensar los efectos de divisiones de acciones y otros ajustes, actualmente es un promedio escalonado. (iForex, 2019)

**USSPX500:** El índice Standard & Poor's 500 (Standard & Poor's 500 Index) también conocido como S&P 500 es uno de los índices bursátiles más importantes de Estados Unidos. Al S&P 500 se lo considera el índice más representativo de la situación real del mercado. El índice se basa en la capitalización bursátil de 500 grandes empresas que poseen acciones que cotizan en las bolsas NYSE o NASDAQ, el índice captura aproximadamente el 80% de toda la capitalización de mercado en Estados Unidos. Los componentes del índice S&P 500 y su ponderación son determinados por S&P Dow Jones Indices. Se diferencia de otros índices de mercados financieros de Estados Unidos, tales como el Dow Jones Industrial Average o el índice Nasdaq Composite, en la diversidad de los rubros que lo conforman y en su metodología de ponderación. (Wikipedia, 2019).

**AAPL:** símbolo que se usa en la bolsa, en el NASDAQ, para Apple Computer, Inc. (Miguel A, 2012)

**MSFT:** símbolo que se usa en la bolsa, en el NASDAQ, para Microsoft Computer, Inc.

**WTI:** West Texas Intermediate (WTI) es una corriente de crudo producido en Texas y el sur de Oklahoma que sirve como referencia para fijar el precio de otras corrientes de crudo

# 

# **Referencias**

Mercado de divisas. Recuperado de *Wikipedia, La enciclopedia libre*. Fecha de consulta: abril 17, 2019 from https://es.wikipedia.org/wiki/Mercado\_de\_divisas

fxstreet. (12 de 04 de 2019). fxstreet. Obtenido de fxstreet: https://www.fxstreet.es/rates-charts/gbpcad

iForex. (11 de 04 de 2019). iForex. Obtenido de iForex: https://www.iforex.mx/%C3%ADndices/dow-jones

Miguel A, G. (17 de 09 de 2012). applesfera. Obtenido de applesfera: https://www.applesfera.com/respuestas/alguien-que-me-explique-por-favor-porque-aapl-es-apple

Perez, A. B. (24 de 10 de 2018). Enciclopedia financiera. Obtenido de https://www.enciclopediafinanciera.com/mercados-financieros/mercados-deacciones.html

tecnicasdetrading. (12 de 04 de 2019). tecnicasdetrading. Obtenido de tecnicasdetrading: https://www.tecnicasdetrading.com/2015/12/par-de-divisasusdjpy.html tecnincasdetrading. (12 de 04 de 2019). tecnincasdetrading. Obtenido de tecnincasdetrading: https://www.tecnicasdetrading.com/2015/11/par-de-divisaseurusd.html

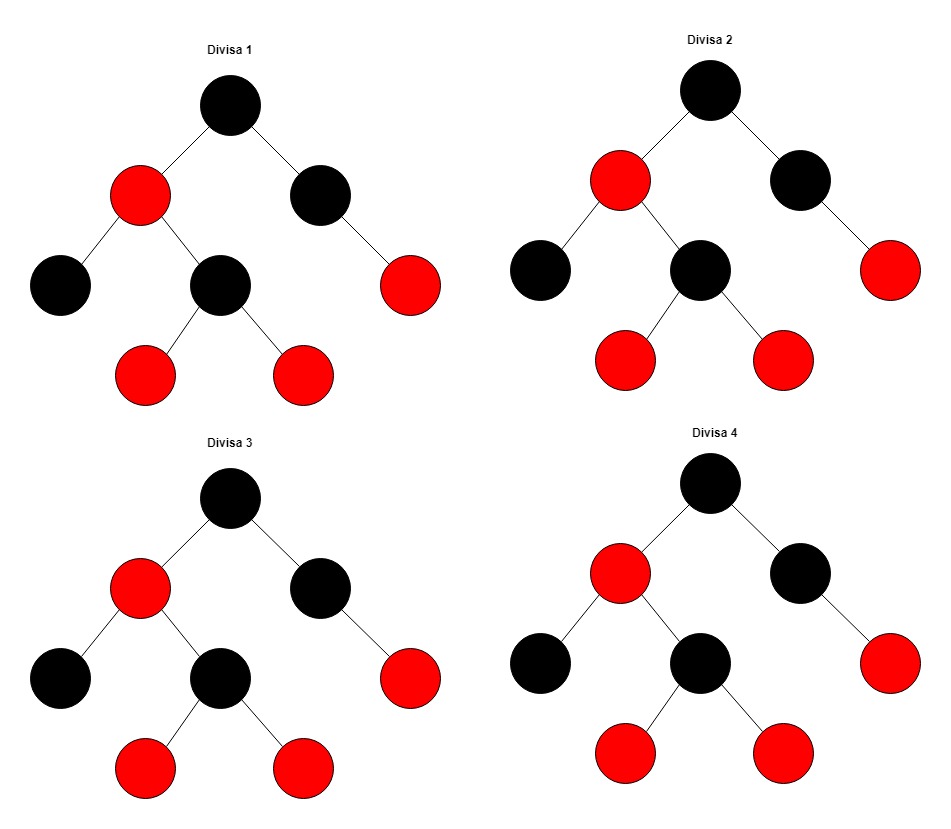
Tradingview. (12 de 04 de 2019). Tradingview. Obtenido de Tradingview: https://es.tradingview.com/symbols/XAUUSD/

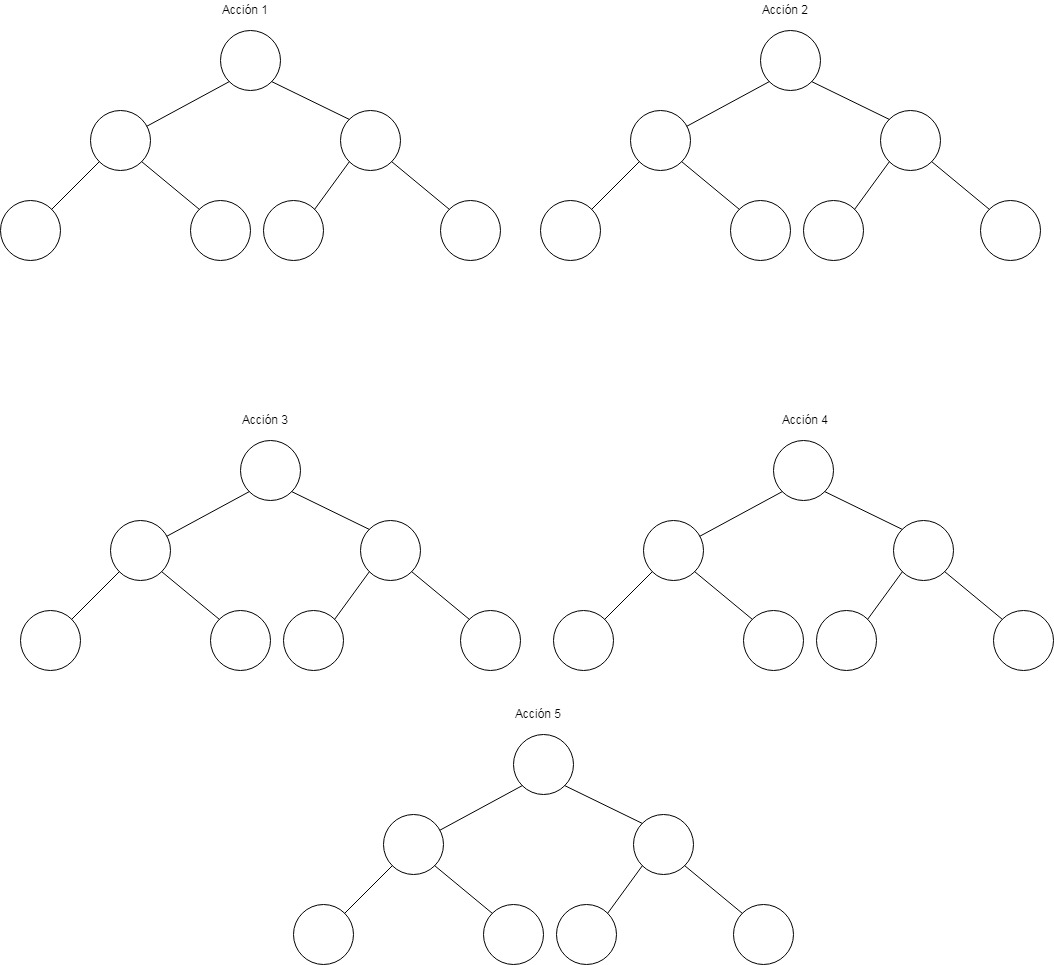
Wikipedia. (03 de 04 de 2019). Wikipedia. Obtenido de Wikipedia: https://es.wikipedia.org/wiki/S%26P\_500

**3. Búsqueda de soluciones creativas**

**Alternativa 1.**

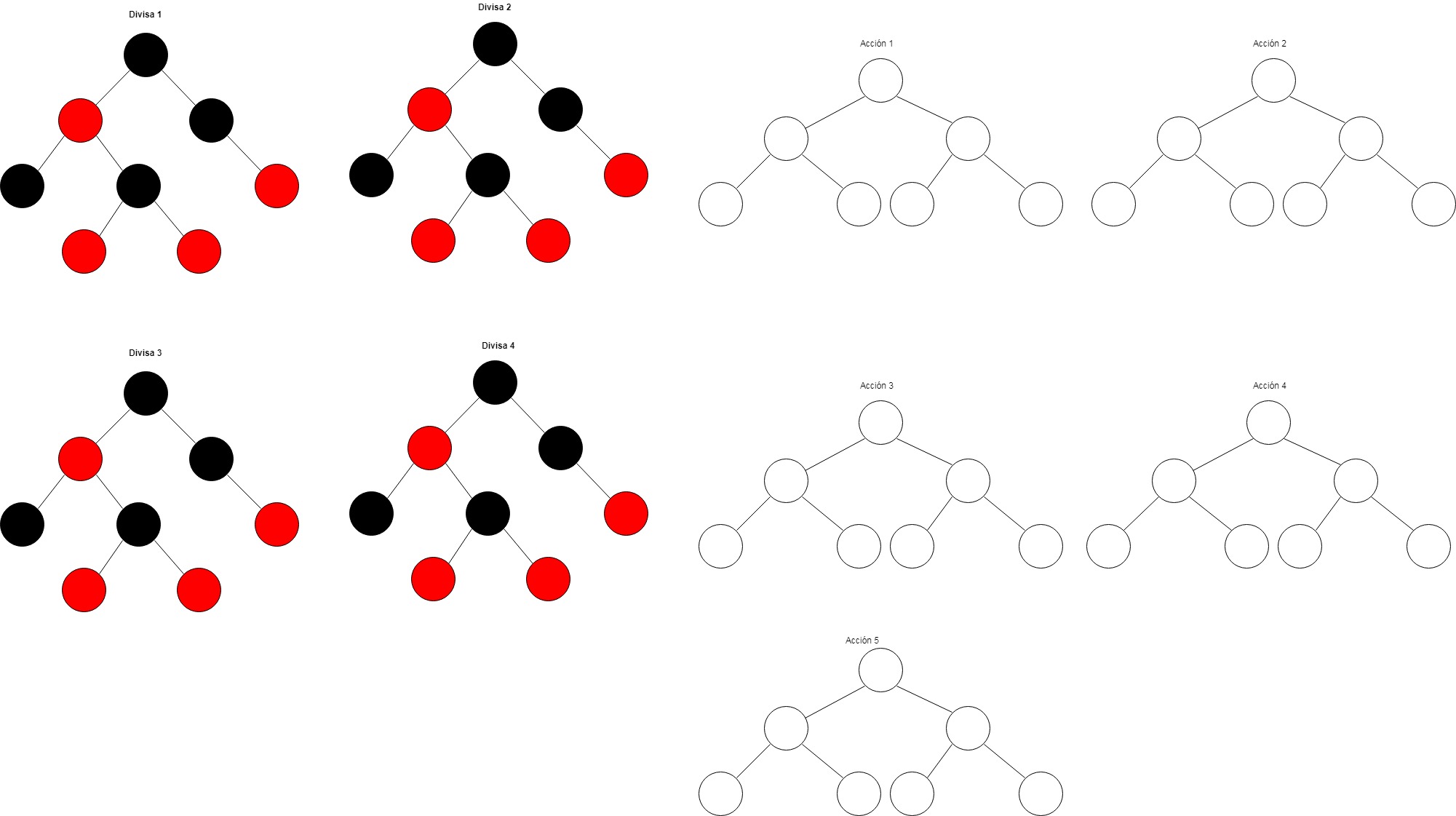
En esta alternativa se utilizará para cada divisa un árbol rojinegro y para cada acción un árbol AVL y se aislaran las divisas de las acciones, es decir, estarán en clases diferentes y por último los bitcoins en un arreglo de tamaño variable.

****

****

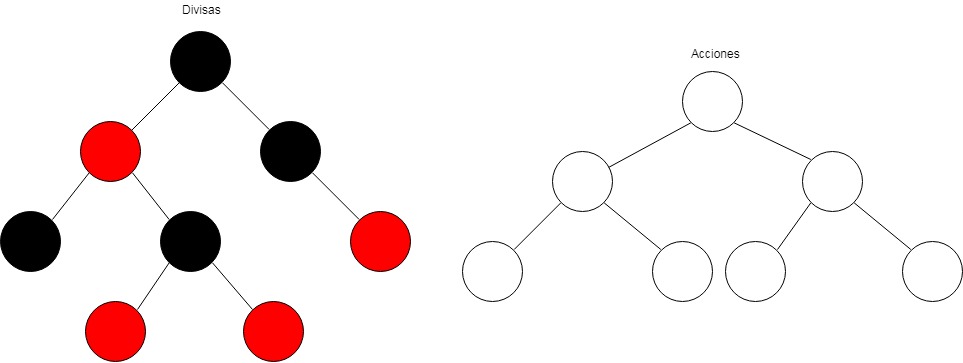
**Alternativa 2.**

para esta alternativa las divisas y las acciones se juntaran, estarán en una misma clase, y cada divisa tendrá un árbol rojinegro diferente y cada acción un árbol AVL diferente y por último los bitcoins en un arreglo de tamaño variable.

****

**Alternativa 3.**

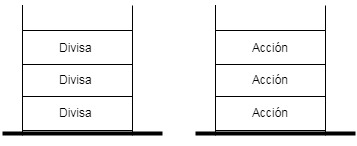
Esta alternativa se basa en que las divisas y las acciones se juntaran, estarán en una misma clase, las divisas tendrá un árbol rojinegro y las acciones un árbol AVL y por último los bitcoins en un arreglo de tamaño variable.



**Alternativa 4.**

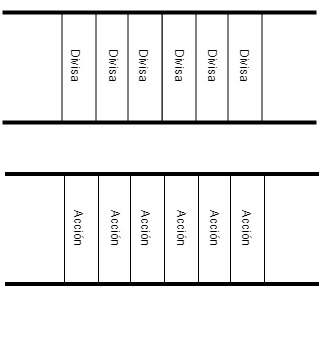
En esta alternativa pondremos todas las acciones y divisas en pilas así podremos obtener los últimos elementos que se agregaron en tiempo constante y por último los bitcoins en un arreglo de tamaño variable.

.



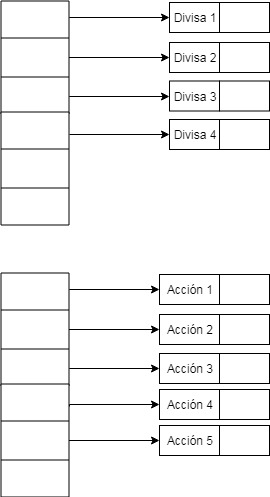
**Alternativa 5.**

En esta alternativa pondremos todas las acciones y divisas en colas así podremos obtener los primeros y últimos elementos que se agregaron en tiempo constante y por último los bitcoins en un arreglo de tamaño variable.



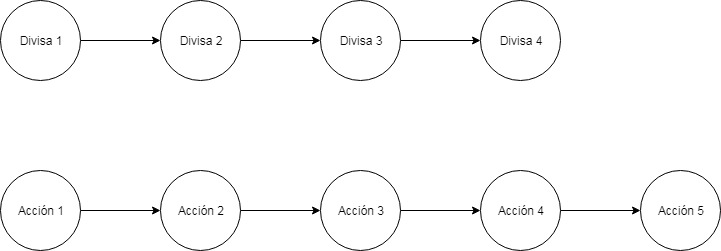
**Alternativa 6.**

Esta alternativa se basa en poner las acciones y divisas en tres tablas hash y así poder acceder cualquier acción, divisa o bitcoin en tiempo constante y por último los bitcoins en un arreglo de tamaño variable.

****

**Alternativa 7.**

En esta última alternativa utilizaremos listas enlazadas para conectar las acciones y divisas y por último los bitcoins en un arreglo de tamaño variable.

****

**4. Diseños preliminares**

La solución de este problema requiere de las estructuras de datos del árbol rojinegro y el AVL, por esta razón y otras a continuación se descartaron las siguientes alternativas de la búsqueda de soluciones creativas:

|  |  |
| --- | --- |
| **Alternativa 4** | Esta alternativa se descarta debido a que almacenar todos los datos en pilas, no resultaría nada eficiente en complejidad espacial y temporal. |
| **Alternativa 5** | 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 tener que eliminar otros haciendo que perdamos información. |
| **Alternativa 6** | Esta alternativa se descarta ya que almacenar todos las divisas y acciones en HashTables es eficiente con respecto al tiempo, no tiene nada que ver con el contexto del problema |
| **Alternativa 2** | Esta alternativa se descarta puesto que es muy similar a la alternativa 1. |

**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 ayudarán 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.

o Excelente: 4 puntos

o Fiel: 3 puntos

o Regular: 2 puntos

o 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.

o Constante: 6 puntos

o Logarítmica: 5 puntos

o Lineal: 4 puntos

o Polinomial: 3 puntos

o Exponencial: 2 puntos

o 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.

o Constante: 6 puntos

o Logarítmica: 5 puntos

o Lineal: 4 puntos

o Polinomial: 3 puntos

o Exponencial: 2 puntos

o 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.

o Constante: 6 puntos

o Logarítmica: 5 puntos

o Lineal: 4 puntos

o Polinomial: 3 puntos

o Exponencial: 2 puntos

o 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.

o Constante: 6 puntos

o Logarítmica: 5 puntos

o Lineal: 4 puntos

o Polinomial: 3 puntos

o Exponencial: 2 puntos

o Factorial: 1 punto

**Evaluación:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Alternativa** | **Criterio A** | **Criterio B** | **Criterio C** | **Criterio D** | **Criterio E** | **Total** |
| **1** | **4** | **4** | **5** | **4** | **4** | **21** |
| **3** | **3** | **4** | **5** | **4** | **4** | **20** |
| **7** | **2** | **4** | **4** | **4** | **4** | **18** |

Con base en los resultados obtenidos la alternativa 2 y 7 van a ser descarta y por lo tanto la alternativa 1 será la utilizada para darle solución al problema.

**6. Preparación de informes y especificaciones.**

* **Diseño de pruebas unitarias**
* **Estructura de Datos: AvlTree**

|  |  |  |  |
| --- | --- | --- | --- |
| Clase: AVLTree | | Método: search | |
| Caso# | Descripción de la prueba | Estado inicial: | Resultado |
| 1 | Se busca un elemento en el árbol AVL. Para este caso se desea buscar los elementos 100, 1 y 55 que se supone deberían estar en el árbol | Un árbol previamente creado con los siguientes valores={5,4,1,8,23,10,0,  7,12,1000} | La prueba se ejecuta correctamente y los valores son encontrados en el árbol. |

|  |  |  |  |
| --- | --- | --- | --- |
| Clase: AVLTree | | Método: add | |
| Caso# | Descripción de la prueba | Estado inicial: | Resultado |
| 2 | Al árbol se agregan los valores que están creados en el stage2. Luego se crea un arreglo con los valores agregados al árbol, de tal manera que correspondan a los valores almacenados en el árbol recorrido por medio del recorrido inorden. | Un árbol AVL previamente creado, así como unas variables de tipo int que representan ciertos valores.(uno, dos, tres cuatro, cinco, diez; que efectivamente representan sus respectivos números) | La prueba se ejecuta sin problemas y efectivamente los dos arreglos(el creado y el devuelto por el método getDFS recorriendo el árbol por medio de inorden) son iguales. |

|  |  |  |  |
| --- | --- | --- | --- |
| Clase: AVLTree | | Método: remove | |
| Caso# | Descripción de la prueba | Estado inicial: | Resultado |
| 3 | Al árbol se agregan los valores que están creados en el stage2.Luego se procede a eliminar el “cinco” por medio del método remove. Y se crea un arreglo con los mismos valores agregados de tal manera que correspondan con el arreglo ideal que debería retornar cuando se recorra el árbol por medio de preorden. | Un árbol AVL previamente creado, así como unas variables de tipo int que representan ciertos valores. (uno, dos, tres cuatro, cinco, diez; que efectivamente representan sus respectivos números) | La prueba se ejecuta sin problemas y efectivamente los dos arreglos(el creado y el devuelto por el método getDFS recorriendo el árbol por medio de preorden) son iguales. |

|  |  |  |  |
| --- | --- | --- | --- |
| Clase: AVLTree | | Método: clear | |
| Caso# | Descripción de la prueba | Estado inicial: | Resultado |
| 4 | Se prueba que el método clear efectivamente funciona, al comparar el tamaño del arreglo que representa el árbol después de que se ejecuta el método, con 0 | Un árbol previamente creado con los siguientes valores={5,4,1,8,23,10,0,  7,12,1000} | La prueba se ejecuta sin problemas, y efectivamente el tamaño del arreglo es 0, asegurando asi que la prueba funciona sin problemas. |

* **Estructura de Datos: RedBlackTree**

|  |  |  |  |
| --- | --- | --- | --- |
| Clase: RedBlackTree | | Método: search | |
| Caso# | Descripción de la prueba | Estado inicial: | Resultado |
| 1 | Se busca un elemento en el árbol rojinegro. Para este caso se desea buscar los elementos 233 y 734 que se supone deberían estar en el árbol | Un árbol previamente creado con los siguientes valores={8,4,34,98,233,1450,0,734,  1200,1000,98765,98754} | La prueba se ejecuta correctamente y los valores son encontrados en el árbol. |

|  |  |  |  |
| --- | --- | --- | --- |
| Clase: RedBlackTree | | Método: add | |
| Caso# | Descripción de la prueba | Estado inicial: | Resultado |
| 2 | Se compara la cantidad exacta de elementos que deberían estar agregadas en el árbol con la cantidad real que retorna el método size() | Un árbol previamente creado con los siguientes valores={8,4,34,98,233,1450,0,734,  1200,1000,98765,98754} | La prueba se ejecuta correctamente y los valores coinciden, demostrando así que el agregar funciona correctamente. |

|  |  |  |  |
| --- | --- | --- | --- |
| Clase: RedBlackTree | | Método: remove | |
| Caso# | Descripción de la prueba | Estado inicial: | Resultado |
| 3 | Se eliminan tres elementos del árbol y luego se procede a comparar si el tamaño de este coincide con el tamaño que debería tomar después de haber eliminado esos elementos, así como a comprobar que cuando se buscan estos elementos en el árbol se retorna null. | Un árbol previamente creado con los siguientes valores={8,4,34,98,233,1450,0,734,  1200,1000,98765,98754} | La prueba se ejecuta correctamente y el tamaño deseado coincide con el tamaño real, también cada prueba retorna su respectivo null cuando se buscan los elementos eliminados, comprobando así que el método remove funciona correctamente. |

* **Diagrama de clases de la solución.**

**7. Implementación de la solución.**

[**https://github.com/MrAlien98/Lab3**](https://github.com/MrAlien98/Lab3)