**Informe: Trabajo 2 Primera parte Bases de datos 2 2021-1**

**Integrantes:**

Santiago Espinosa Arteaga cc 1037671102 Juan Pablo Carvajal cc 1152472428

**Punto 1**

**Nota:** Por favor ejecutar los comandos en una terminal CMD o Bash ya que powershell pone problemas con las comillas

En primera instancia se crearon los comandos que permitieran crear la blockchain “empresa”, los streams de vendedores y ganancias.

Después de esto aparte se crearon los smart filters, los cuales funcionan de la siguiente manera:

**Smart filter de vendedores:**

En primer lugar se captura el objeto que se desee insertar al cual llamamos **obj** , luego de esto se evalúa la propiedad **obj.data.format** con el formato JSON, en caso de no serlo se rechaza la inserción. Después de esto se guarda el json que trae **obj** mediante una variable **json** que es igual a **obj.data.json.** Sin embargo antes de continuar se comprueba que **obj.keys.length** (Que es igual a la longitud de claves de **obj**) sea igual a 1, en caso de no serlo se rechaza la inserción. Una vez aprobado esto mediante el método **json.hasOwnProperty** se comprueba primero que el json tenga el atributo **nombre** y después **teléfono**, en caso de no tenerlo se rechaza la inserción. Por último mediante la propiedad del objeto **Object, Object.keys(json).length** (La cual evalúa la cantidad de propiedades del json a insertar)se comprueba que el json solo tenga 2 propiedades, en caso de no serlo se rechaza la petición, de lo contrario la inserción pasaría este smart filter

**Smart filter de ganancias**

Este smart filter funciona de una manera muy similar al de vendedores, en primera instancia se comprueba que el formato del objeto a insertar **obj** sea un JSON, después se comprueba que la cantidad de claves de este json sean 1, en caso de no pasar cualquiera de los filtros anteriores se rechaza la inserción. Posteriormente se guarda el json a insertar en una variable llamada **json,** a esta variable se le comprueba que tenga la propiedad **valor** de la misma forma que se comprobó con el teléfono y el nombre en el smart filter de vendedores. Por último mediante el método **Object.keys(json).length** que **json** solo tenga 1 propiedad, en caso de no pasar alguna de las verificaciones anteriores se rechaza la inserción, de lo contrario se deja seguir.

**Punto 2**

Al dar click en el botón registrar ganancia se creará una instancia de la clase registroGanancia, ésta además de los elementos gráficos básicos y etiquetas incluye un campo de texto en el que se recibe el valor de la ganancia y se genera una lista con las cédulas, para la creación de esta lista se crea una instancia de la clase ApiMultichainBD2 en la cual se genera la conexión a la blockchain y se encuentran los métodos para interactuar con esta, los parámetros de la instancia son el servidor, el default-rpc-port, el rpcuser y la rpcpassword, luego de esto se crea la variable vendedores la cual contendrá las cédulas, es decir las keys de los ítems de los streams a través del método obtenerCedulasVendedores() de la clase ApiMultichainBD2, este método se subscribe al stream vendedores y guarda en una lista “items” todos los ítems presentes en dicho stream, luego itera sobre los elementos de “ítems” y agrega cada key a una lista enlazada resultado, finalmente devuelve esta lista enlazada. Luego de tener la lista vendedores, se crea el array arrayCedulas con estos datos ya que el formato que recibe el objeto lista de la interfaz gráfica es de tipo array, luego se realiza un control para manejar datos con keys vacías ya que la interfaz no los muestra correctamente y finalmente arrayCedulas es enviado a la lista de la interfaz. Al dar click en el botón ingresar se invocará la función insertarGanancias a través de la instancia de la clase ApiMultichainBD2 con los parámetros ingresados en la caja de texto y en la lista, esta función convierte el valor de la ganancia a un string en forma de json con el valor enviado a través de la función convertir ganancia, luego realiza un publish al stream ganancias con la cédula enviada como key y el json que contiene el valor y finalmente se es retornado a la ventana de inicio.

Al dar click en el botón registrar vendedor se creará una instancia de la clase registroVendedor que igualmente incluirá elementos gráficos básicos y etiquetas junto con 3 campos de texto, nombre, cédula y teléfono y se creará una instancia de la clase ApiMultichainBD2 de igual forma que en la ventana de registrar ganancia. Al dar click al botón de ingresar se llamará al método insertarVendedor de la clase ApiMultichainBD2 con los parámetros insertados en las cajas de texto, ésta primero llama a la función convertirVendedor la cual toma los parámetros nombre y teléfono y los retorna un string con estructura de json con dichos valores, luego de esto se realiza un publish al stream vendedores con la cédula como key y el json retornado de convertirVendedor y se retorna a la ventana de inicio.

Al dar click en consultar se creará una instancia de la clase consulta la cual además de elementos gráficos básicos y etiquetas, desplegará 2 listas, una con las cédulas de los vendedores y otra con sus respectivas ganancias totales, para la creación de la lista de las cédulas se sigue el mismo que en la ventana de registrar ganancias, para la lista de ganancias se realiza un ciclo con i iniciando en cero e incrementando en 1 su cantidad en cada iteración mientras este sea menor que el tamaño de la lista con las cédulas, durante este ciclo se asignará en la posición i del arreglo arrayGanancias el resultado de la función obtenerGananciasPorVendedor al enviar el valor de la lista de cédulas en la posición i, es decir que cada posición del arreglo de ganancias contendrá el valor de la función obtenerGananciasPorVendedor de cada cédula del arreglo de cédulas, la función obtenerGananciasPorVendedor de la clase ApiMultichainBD2 recibe un string y crea una variable Double llamada ganancia con valor 0.0, seguido de esto se subscribe al stream ganancias y guarda en una lista “items” todas los ítems del stream ganancias cuya key sea la el parámetro enviado al llamar la función, luego de esto se itera sobre los elementos de “items” y se extrae el valor del atributo “valor” del json de cada item, en cada iteración este valor es sumado a la variable valor creada al inicio del método con lo que al final se obtendrá la suma de todos los valores de las ganancias cuya key corresponda al parámetro cédula, de esta forma se desplegarán ambas listas, una mostrando las cédulas de los vendedores del stream vendedores y otra con la suma de las ganancias asociadas a cada cédula.