**Parte 1 del Trabajo 2 (40%) sobre Multichain**

**Nota: La parte 2 (60%) sobre Solidity se publicará luego de la entrega de la parte 1.**

**Enunciado realizado por Daniel Sánchez, revisado por Francisco Moreno**

**Punto 1 (20%)**

Cree una *blockchain* llamada “empresa”. En esta *blockchain*, cree un *stream* llamado “vendedores” y un *stream* llamado “ganancias”.

Desarrolle un *smart filter* para *stream* que controle lo siguiente:

Cuando se vaya a insertar un elemento en el *stream* “vendedores”, este elemento:

* Debe tener una sola clave.
* Debe ser de tipo JSON.
* Debe tener los siguientes campos (los nombres de los campos deben ser exactamente iguales (sin tildes) y NO debe tener más o menos campos):
* nombre
* telefono

Desarrolle un *smart filter* para *stream* que controle lo siguiente:

Cuando se vaya a insertar un elemento en el *stream* “ganancias”, este elemento:

* Debe tener una sola clave.
* Debe ser de tipo JSON.
* Debe tener el siguiente campo (el nombre del campo debe ser exactamente igual y NO debe tener más o menos campos):
* valor

**Punto 2 (20%)**

Desarrolle una interfaz gráfica en Java para lo siguiente.

Al iniciar el programa, el usuario debe ver una ventana con tres botones: “Consultar”, “Registrar vendedor” y “Registrar ganancia”.

Al dar clic en el botón “Registrar vendedor” debe llevar al usuario a una ventana en la que se pueda ingresar un vendedor a la *blockchain*. Para ello, la ventana debe tener tres (3) campos de texto: uno para la cédula, uno para el nombre y uno para el teléfono. Además, esta ventana debe tener un botón “**Ingresar**”; al darle clic a este botón, se debe insertar el vendedor como un JSON con el formato descrito en el punto anterior en el *stream* “vendedores” de la *blockchain* “empresa”. **La clave de este JSON debe ser la cédula ingresada**.

Al dar clic en el botón “Registrar ganancia” debe llevar al usuario a una ventana en la que se pueda ingresar una ganancia a la *blockchain*. Para ello, la ventana debe tener dos (2) campos: una lista desplegable que permita seleccionar la cédula del vendedor de entre las cédulas que ya existan en el sistema y un campo de texto para ingresar el valor de la ganancia. Además, debe tener un botón “**Ingresar**”; al darle clic a este botón, se debe insertar la ganancia como un JSON con el formato descrito en el punto anterior en el *stream* “ganancias” de la *blockchain* “empresa”. La clave de este JSON debe ser la cédula seleccionada.

Al dar clic en el botón “Consultar” se debe mostrar la cédula de cada vendedor y la **suma** total de los valores de sus ganancias.

**Notas adicionales:**

* Puede usar todas las estructuras de datos que desee.
* **Se garantiza que en las pruebas solo se ingresarán datos correctos, por ejemplo, en el campo valor se ingresarán solo números enteros positivos correctos.**
* Para entregar por email a *[fjmoreno@unal.edu.co](mailto:fjmoreno@unal.edu.co)*, el miércoles 21 de abril **hasta las 6 pm**. Solo se califican trabajos enviados a ese correo.
* **No se reciben trabajos en hora posterior**. No se reciben versiones “mejoradas”. No se califican trabajos enviados “por accidente” a otros correos.
* **Se debe incluir un informe breve donde se describa cómo se solucionó cada punto.** Este informe hace parte de la calificación del trabajo (máximo 2 hojas). **No enviar los datos de prueba que usted usó para probar sus códigos.**
* Grupos de **tres** personas.
* Los trabajos deben ser independientes entre los grupos. Trabajos copiados **así sea en un SOLO punto** se califican con 0 (cero) en su totalidad para todos los integrantes. Las soluciones presentadas deben ser originales. El trabajo debe ser desarrollado por los integrantes del grupo no por personas ajenas a él.
* El monitor les puede ayudar con aspectos técnicos pero su función **no** es hacerles la práctica **ni está autorizado** para **cambiar las condiciones del trabajo**. Sin embargo, en este trabajo él si tiene la potestad de **aclarar** dudas sobre las condiciones del Multichain.
* **Si trabaja con otras herramientas**, así su trabajo funcione y sea “espectacular”, el trabajo **NO** será calificado.
* **No** se califica la “belleza” de los formularios, se califica la funcionalidad.
* Cualquier duda consultarla con el profesor o con el monitor.

Abril 9 de 2021