**Punto 1 del Trabajo 2 (50%)**

**Nota: El Punto 2 (50%) del Trabajo 2 será sobre Solidity y se publicará luego de entregar el Punto 1.**

**Trabajo Multichain – *Art Blockchain***

**a)  
(10%)** Cree una *blockchain* en Multichain llamada “Art” en un equipo (puede ser una máquina virtual o un computador de uno de los miembros del grupo). Este nodo será el nodo admin de la *blockchain*.Una vez creada la *blockchain*, conecte otro equipo como un segundo nodo a la *blockchain* “Art”.Cree un *stream* llamado “Obras” y otro llamado “Autores”. Estos *streams* se crean desde el nodo admin, pero se manipularán desde el segundo nodo, asegúrese de **conceder los permisos** necesarios para esto.

**(30%)** Desde el nodo admin cree los siguientes *smart filters*:

* *Smart filter* de *stream* que valide lo siguiente para el *stream* “Autores”:
  + Debe tener una única *key*. Esta *key* será el *nickname* con el que se identificará a cada autor.
  + Solo se permiten datos tipo JSON.
  + El JSON debe contener los siguientes campos (ni más, ni menos): *nombre, agno\_nacimiento* y *correo.* Queda a su elección validar: que el campo agno\_nacimiento sea numérico y que el campo correo sea una dirección de correo válida.
* *Smart filter* de *stream* que valide lo siguiente para el *stream* “Obras”:
  + Debe tener exactamente dos *keys*. Estas *keys* corresponden al autor (*nickname*) de la obra y al nombre que la identifica (la primera *key* corresponde al autor y la segunda al nombre de la obra).
  + Solo se permiten datos tipo JSON.
  + El JSON debe contener los siguientes campos (ni más, ni menos): *fecha\_creacion* y *obra* (es decir, la imagen, ver punto b) donde se dan más detalles sobre este campo).Queda a su elección validar que el campo fecha\_creación sea una fecha válida.

Estos filtros deben ser activados solo para el segundo nodo.

Para este punto, envíe un archivo de texto con los comandos que usó para hacer lo solicitado indicando claramente en qué nodo se ejecuta cada comando y el orden en el que se deben ejecutar. Para los *smart filters*, incluya por separado el código de la función de forma legible (archivo txt tal que al copiar y pegar no haya caracteres extraños que generen errores de compilación).

**b)**Cree un programa en Java que permita interactuar con la *blockchain* creada en el punto anterior por medio de una interfaz gráfica. Dicho programa se deberá conectar a la *blockchain* desde el segundo nodo y deberá contar con las siguientes funcionalidades:

* **(20%) Registro autor***:* Permite ingresar los campos necesarios para almacenar un autor en el *stream* “Autores”, incluyendo el *nickname* que se usará como *key*.
* **(20%)*****Subir obra****:* Permite ingresar los campos necesarios para almacenar una obra en el *stream* “Obras”, incluyendo sus dos *keys*, donde la *key* que hace referencia al autor debe corresponder a un *nickname* (*key*) del *stream* “Autores” (el programa Java debe hacer la validación correspondiente).   
  Para ingresar las imágenes en la *blockchain* (campo *obra* del JSON), se recomienda convertir dichas imágenes a un formato de texto usando, por ejemplo, esta herramienta: [Image to Base64 converter.](https://codebeautify.org/image-to-base64-converter) También puede implementar un codificador y decodificador de imágenes en su programa si así lo desea. Se garantiza que solo se ingresarán imágenes con un tamaño inferior a 10KB.
* **(20%)*****Ver obras****:* Permite listar las obras filtrando por autor (*nickname*) o por nombre de la obra (o por ambos filtros a la vez) junto con la información almacenada de dicha obra y su autor. Se da libertad para decidir la forma en la que se muestra el campo “obra” del *stream* “Obras”, lo ideal es mostrar la imagen original. **Nota**: Si en la *blockchain* hay varios autores que tienen el mismo *nickname* o varias obras que tienen el mismo nombre, se deja a su elección como proceder en estos casos.

Para este punto, envíe el proyecto Java SIN construir, de modo que se puedan modificar los parámetros de la conexión.

**Notas adicionales:**

* Para enviar por email a [*fjmoreno@unal.edu.co*](mailto:fjmoreno@unal.edu.co) y a *jpenat@unal.edu.co* el lunes 10 de octubre **hasta las 4 pm**. Solo se califican trabajos enviados a esos dos correos.
* **No se reciben trabajos en hora posterior**. No se reciben versiones “mejoradas”. No se califican trabajos enviados “por accidente” a otros correos.
* Grupos máximo de **tres** personas.
* Los trabajos deben ser independientes entre los grupos. Trabajos copiados **así sea en una SOLA parte** se califican con 0 (cero) en su totalidad para todos los integrantes.
* El monitor les puede ayudar con los aspectos técnicos, **en especial de instalación y explicación de comandos**, pero su función **no** es hacerle el trabajo **ni está autorizado** para **cambiar las condiciones del trabajo, pero sí les puede hacer aclaraciones con respecto al enunciado ya que en gran parte fue elaborado por él.**
* **Si trabaja con otros lenguajes o herramientas**, así su trabajo funcione y sea “espectacular”, el trabajo se califica con cero.
* Si encuentra errores en el enunciado por favor avisar lo más pronto posible para hacer las correcciones respectivas.
* Favor enviar en un solo mensaje todo el trabajo. Incluya el nombre de todos los integrantes del grupo en el archivo txt de los *smart filters*.

**Agradecimientos al monitor por la elaboración del enunciado;**

**Revisado y modificado levemente por Francisco Moreno** el 26 de **septiembre de 2022.**