

Servicio de telemedicina

Recreación de un servicio de gestión de pólizas de seguros empleando tecnología blockchain. La blockchain utilizada será del tipo Ethereum donde se podrán desplegar smart contracts para hacer las distintas gestiones de la aseguradora. Los smart contracts estarán programados en Solidity. De este modo, toda la actividad sobre la prestación de los servicios será públicamente auditable:

- listado de asegurados
- listado de servicios
- compra de derechos de servicios (tokens) por parte de los asegurados
- peticiones de prestación de servicios
- altas y bajas de pólizas

Por ello la Compañía de Seguros debe hacer un despliegue de un Smart Contract **InsuranceFactory (IF)** que permita hacer toda la gestión de esta práctica. A través de este Smart Contract los asegurados podrán comprar tokens que cumplen con el estándar **token ERC 20** para poder hacer el pago de los servicios. Los clientes tendrán que comprar los tokens con criptomonedas (ethers). Como es natural, la compañía deberá generar y tener en su balance una cantidad de tokens suficientes para poder venderlos a los asegurados. También, a través del Smart Contract IF de la compañía se han de poder dar de alta los distintos servicios con el precio correspondiente expresado en tokens.

Para poder ser un asegurado, cada cliente de la compañía tiene que ejecutar una función para crear su Smart Contract la **InsuranceHealthRecord (IHS)**. Este IHS servirá para:

- guardar los tokens que compre el usuario
- solicitar los servicios y pagar por ellos
- consultar su historial
- darse de baja como asegurado de la compañía (la compañía tiene que hacer un redeem los tokens)

Los Smart Contracts emitirán eventos de cada una de las acciones que suceden durante la gestión del servicio para que los distintos actores de esta práctica puedan estar al corriente y tengan facilidad de saber los hechos que ocurren.

El proyecto va a incorporar la gestión de los laboratorios (las entidades que son los proveedores de los servicios): inicialmente, la solicitud y la prestación del servicio se puede hacer a través de la compañía. Ahora bien, de forma adicional puede hacer que un proveedor de servicios (por ejemplo, un laboratorio) se pueda dar de alta como nuevo

servicio de la compañía de manera similar a como lo hacen los clientes. De este modo, el Smart Contract del proveedor del servicio gestionará los servicios que dé, así como los cobros que realice para la prestación del mismo.

Aspectos técnicos:

La compañía (Smart Contract IF) deberá tener estructuras de datos para guardar los datos de cada asegurado y de cada servicio. Se deberán combinar, por tanto, las structs, los arrays y / o los mappings para guardar esta información.

La compañía deberá tener funciones de creación, venta y recompra de tokens. Estas funciones deben poder recibir o pagar ethers. También se ha de poder consultar el saldo de tokens de la compañía. También serán necesarias todas las funciones del tipo getters para poder consultar los datos de la compañía, los asegurados que tiene y los servicios.

A cada póliza, la compañía debe permitir al asegurado a través de una función crear un Smart Contract para gestionar su póliza. También el asegurado debe poder dar de baja y la compañía le debe devolver el dinero correspondiente a los tokens que tiene.

Adicionalmente, a través de la compañía se han de poder consultar cuáles son las direcciones de cada póliza y cuál es la dirección de su asegurado.

Finalmente, a través de la aseguradora deben poder consultar los datos de un servicio para poder pedir su prestación. En función de si se opta por la gestión de proveedores de servicios, se deberán añadir a la compañía las funciones de prestación de los servicios o no.

Respecto al Smart Contract de la póliza del cliente (IHS) debe tener estructuras de datos para guardar la información de su seguro y de cada uno de los servicios que ha solicitado. Igualmente que en el caso de la aseguradora deberán combinarse, por tanto, las structs, los arrays y / o los mappings.

Cada asegurado, a través del Smart Contract, tendrá una función Payable para comprar tokens. También una función para solicitar un servicio y hacer la transferencia de tokens correspondiente. Como es natural, se deben poner todas las funciones del tipo getters para poder acceder a las informaciones necesarias de cada asegurador. Finalmente debe haber una función con un selfdestruct () para darse de baja del servicio.