



UNIVERSIDAD DE BUENOS AIRES

FACULTAD DE INGENIERÍA

2DO CUATRIMESTRE DE 2022

75.10 TÉCNICAS DE DISEÑO

Crypto Monitor

GRUPO 07

Integrantes:

Padrón:

Dante Reinaudo <dreinaudo@fi.uba.ar>

102848

Lucia Pardo <lpardo@fi.uba.ar>

99999

Violeta Perez Andrade <viperez@fi.uba.ar>

101456

Tizziana Mazza Reta <tmazzar@fi.uba.ar>

101715

Índice

1. Introducción	2
2. Objetivo	3
3. Diagramas	4
3.1. Modelo de dominio	4
3.2. Arquitectura	5
3.3. Diagramas de casos de uso	6
3.4. Diagramas de clases	7
4. Desarrollo	9
4.1. LogIn	9
4.1.1. LogIn Normal	9
4.1.2. LogIn con Google	9
4.2. Token	11
4.3. Usuarios	12
4.3.1. Usuario común	12
4.3.2. Administrador	13
4.4. Binance	14
5. Conclusión	15

article graphicx float pifont

1. Introducción

En el presente trabajo práctico se realizará un sistema de reglas en base a información sobre las variaciones del mercado en criptomonedas.

El sistema otorgará dos tipos de acceso, uno de administrador con el cual podrá escribir reglas y editarlas y el de usuario normal el cual podrá ingresar a su cuenta y tener un historial de transacciones y el estado de su billetera.

Las reglas se ejecutarán automáticamente según se observen ciertas variaciones en el mercado. Cada regla se compone de una condición y un conjunto de acciones a realizar en caso de que la condición se cumpla. Las acciones posibles son: comprar, vender o modificar una variable. La compra y venta de criptomonedas se realiza a través de un servicio externo de Binance.

Tanto las condiciones como las acciones dependen de distintos valores. Un valor es un resultado único de texto, numérico o booleano que puede obtenerse de diferentes formas: a partir de una variable definida por el usuario, como valor constante, desde la billetera del usuario para una moneda específica o como el resultado de ejecutar una función predefinida.

El sistema cuenta con un frontend que permite la creación de reglas, configurar la autenticación del usuario con Binance y definir variables

2. Objetivo

El objetido de este trabajo se divide en varios ítems que definimos a continuación.

- ⇒ Construir una plataforma que permita monitorear los valores de criptomonedas.
- ⇒ Definir reglas de compra y venta, que apliquen según las condiciones o variaciones del mercado.
- ⇒ Permitir monitorear la variación del valor de una criptomoneda frente a otra.
- ⇒ Permitir crear reglas, que puedan combinarse y que al cumplirse dichas reglas se ejecuten acciones predefinidas de compra o venta.
- ⇒ Contar con un rol administrador que pueda configurar las reglas y las acciones.
- ⇒ Contar con un rol de sólo lectura que pueda entrar y ver el estado de la billetera y las transacciones ejecutadas
- ⇒ Poder ver la billetera, es decir las cantidades de cada moneda, desde ambos usuarios.
- ⇒ Poder ver el historial de las transacciones desde ambos usuarios.

3. Diagramas

3.1. Modelo de dominio

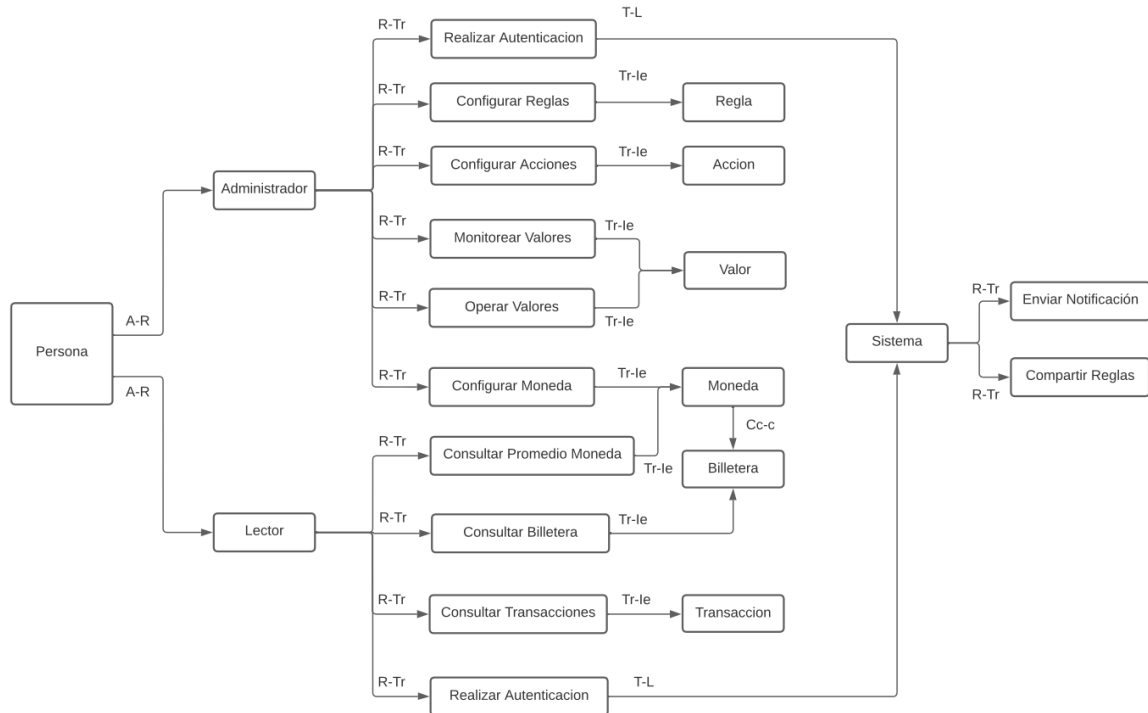


Figura 3.1 – Modelo de dominio del sistema.

Se observa la presencia de dos actores, **Administrador** y **Lector**, y las acciones de los mismos.

3.2. Arquitectura

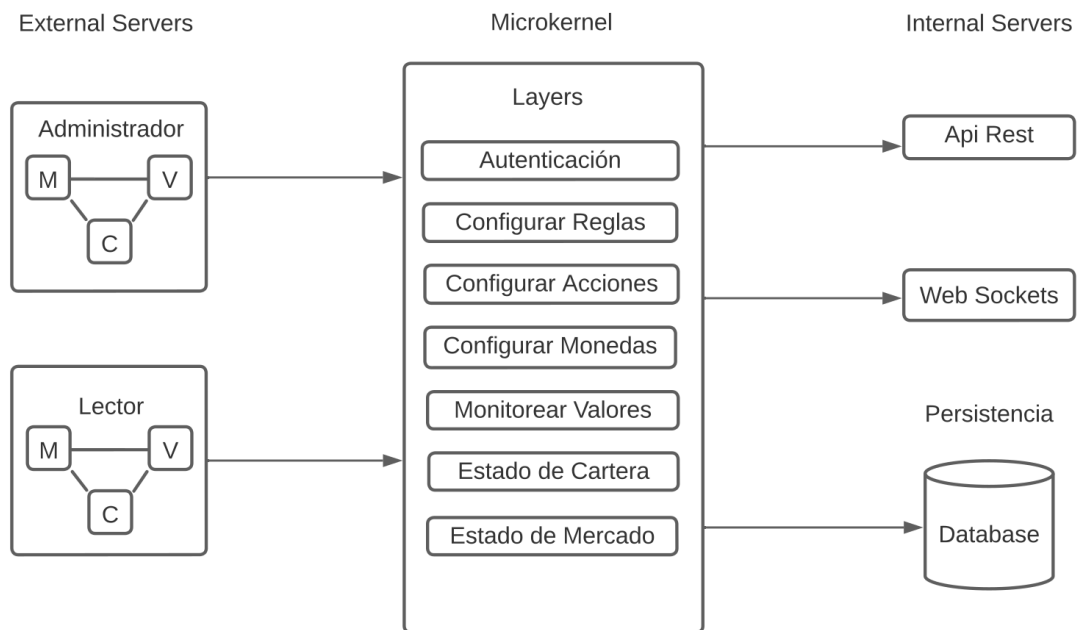


Figura 3.2 – Arquitectura del sistema.

La arquitectura que presentamos es MicroKernel con Layers, en las que se encuentran las distintas acciones que se pueden realizar desde los usuarios.

3.3. Diagramas de casos de uso

A partir de los siguientes diagramas de casos de uso se muestran las acciones que se pueden llevar a cabo, según el tipo de usuario.

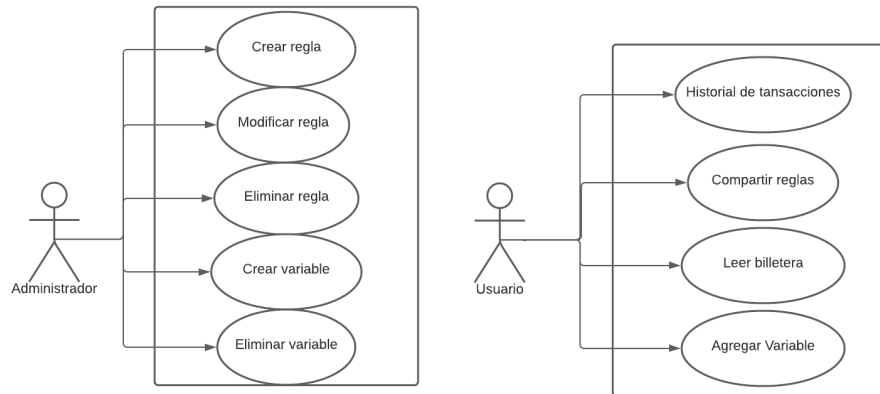


Figura 3.3 – Casos de usos para el usuario como administrador y como usuario común.

El usuario de tipo administrador puede crear, modificar y eliminar reglas, como también crear y eliminar variables, mientras que el usuario común solo puede ver el historial de transacciones, la billetera, compartir reglas y agregar variables al sistema.

3.4. Diagramas de clases

Los diagramas de clases aquí presentados, nos muestran como ha ido evolucionando el trabajo realizado. Se muestran cuales fueron las primeras ideas propuestas.

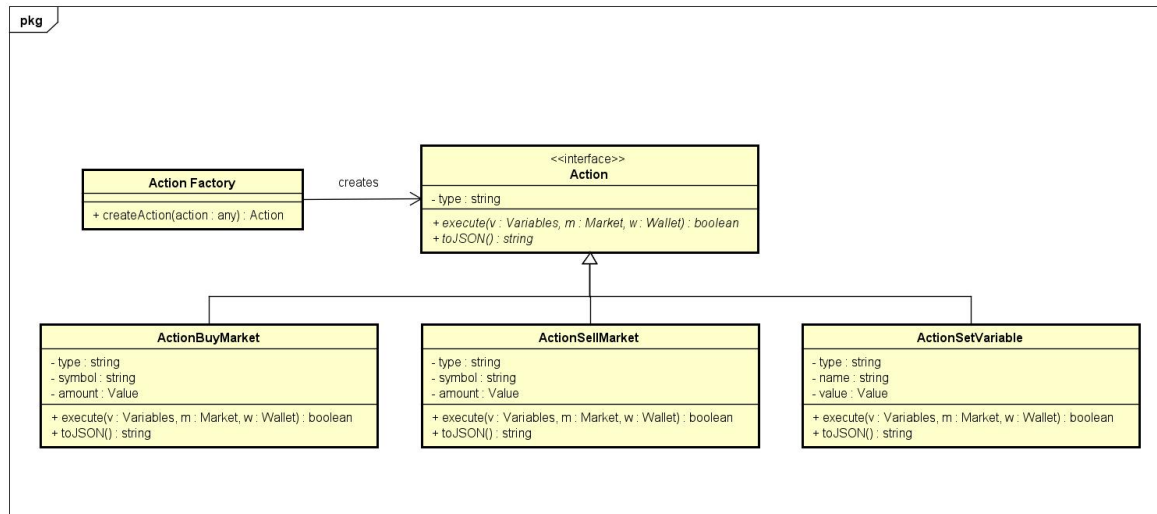


Figura 3.4 – Diagrama de clases de las acciones.

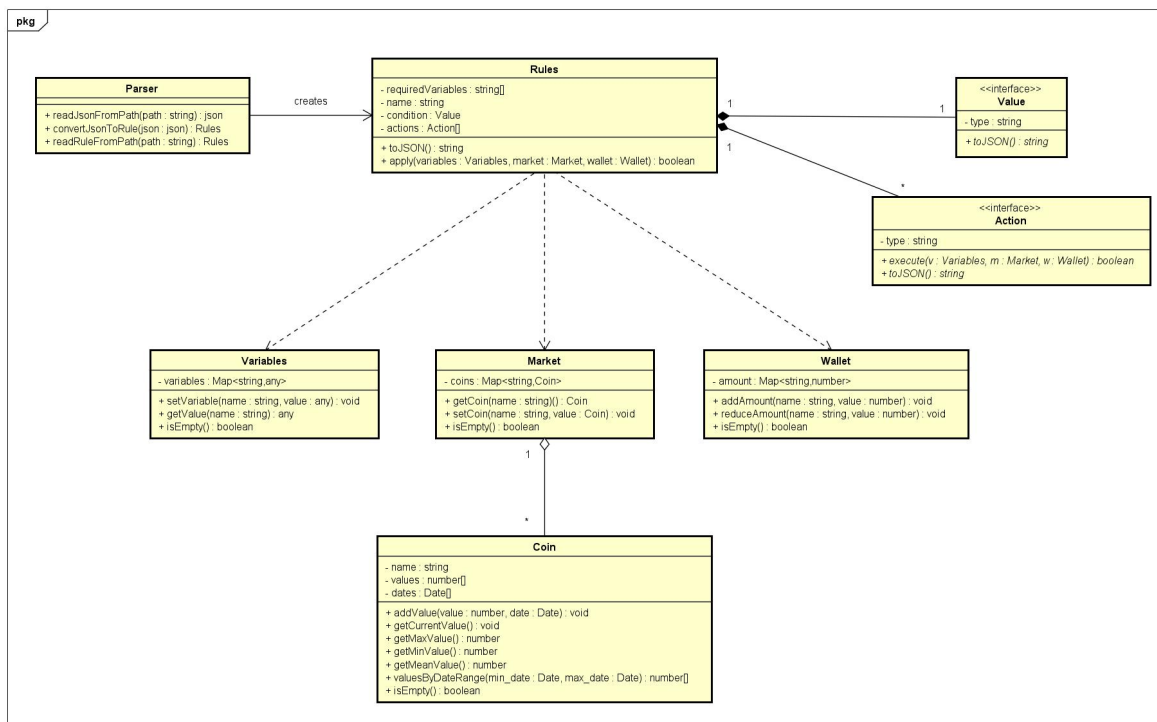


Figura 3.5 – Diagrama de clases de las reglas.

De esta forma fuimos aplicando técnicas de diseño, viendo como era la lógica, y a raíz de la misma, construimos una solución para los problemas dados.

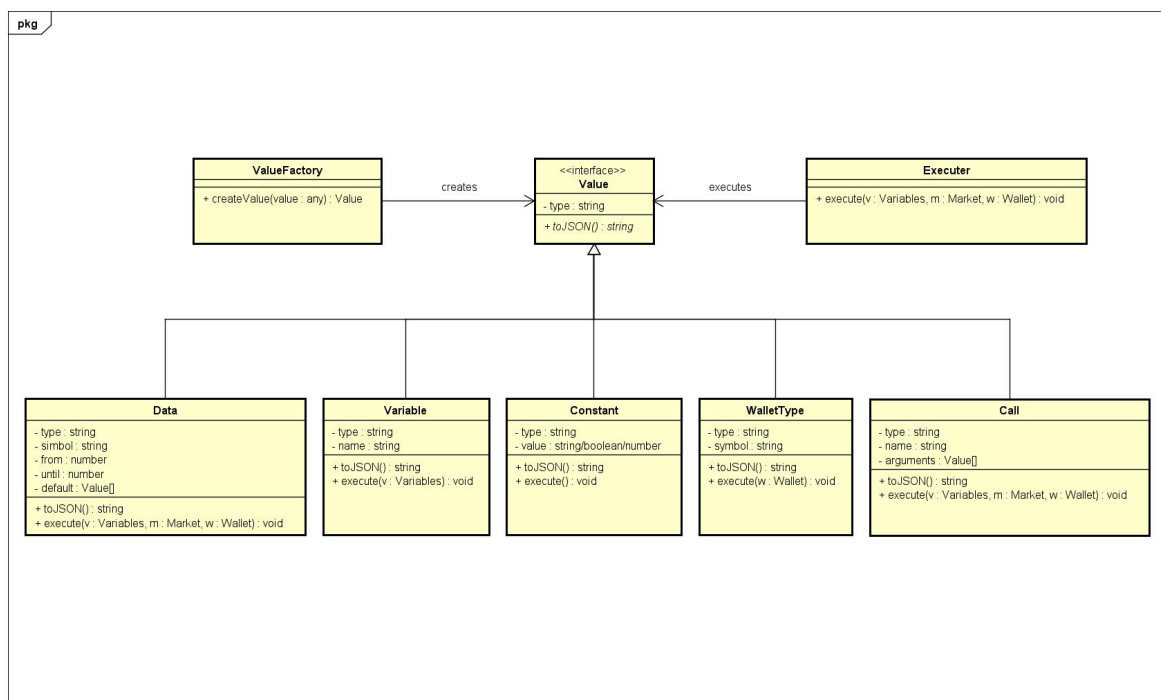


Figura 3.6 – Diagrama de clases de los valores.

4. Desarrollo

4.1. LogIn

Para iniciar sesión en la página de CryptoMonitor, lo primero que debemos hacer es logearnos. Para ello, tenemos dos métodos claves, un login "normal" uno con el mail de Google, que los explicaremos a continuación.

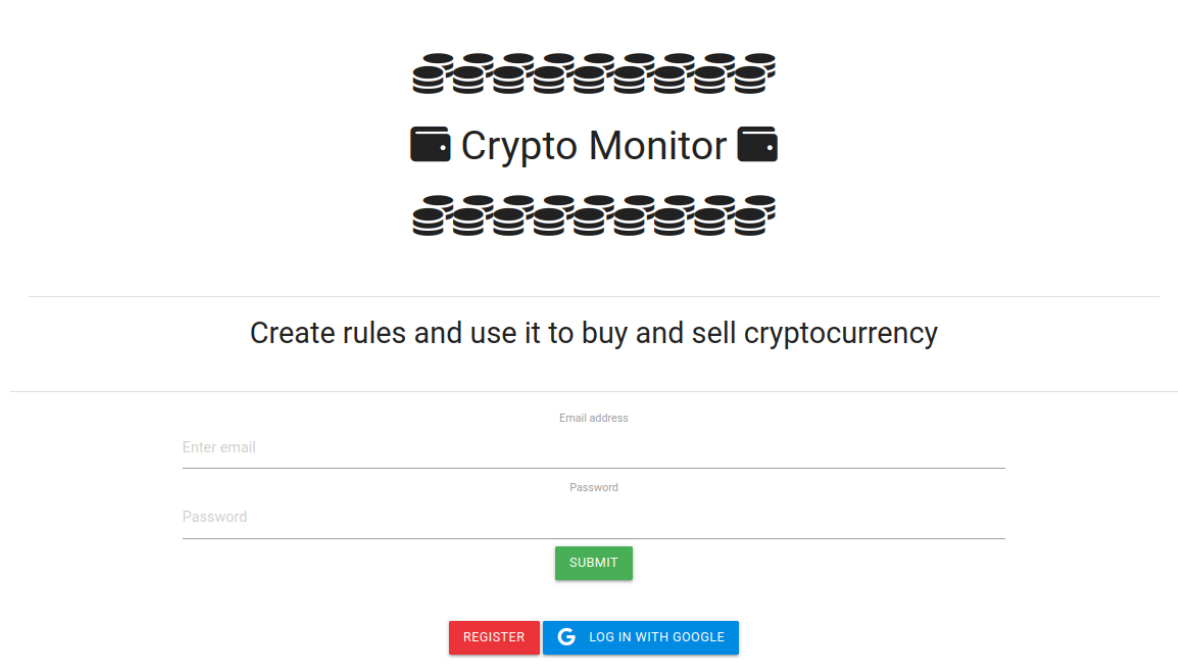


Figura 4.1 – Página principal de CryptoMonitor

4.1.1. LogIn Normal

El login normal es muy sencillo, y lo primero que se debe hacer para tener accesos al mismo, una vez que estamos en la página principal, es apretar el botón rojo que nos aparece en la imagen anterior, en el que dice 'Regiter'. Una vez que accedemos allí, lo próximo que se ve es la sección de registro de un nuevo usuario al sistema, en donde se piden los datos primordiales.

Una vez que se termina de registrar, se submitea el form y, en el caso de que no exista otro usuario con el mismo email utilizado para el registro, se accede a la página de CryptoMonitor. En caso contrario, se vuelve a la página principal del login.

Lo bueno de registrarse de esta manera es que luego, el usuario va a poder ingresar a su cuenta logeandose como se mostro anteriormente, y, además, con el login de Google que vamos a ver a continuación.

4.1.2. LogIn con Google

En este segundo login disponible se tiene una ventaja a la hora del registro, ya que este no es necesario. Teniendo una cuenta de Google, y accediendo con ella, la cuenta se registra automáticamente sin necesidad de utilizar una contraseña.

First name
First name

Last name
Last name

Email address
Enter email

Password
Password

SUBMIT FORM

Figura 4.2 – Casos de usos para el usuario y Administrador

En el caso de haber entrado por primera vez a la pagina registrandose por el método nombrado, lo que sucede es que el usuario siempre deberá logearse de esta forma, ya que, al nunca haberse registrado de la forma convencional, no ingresó ninguna contraseña con la que sea válida entrar por login "normal".

4.2. Token

Para este caso se utilizó JSON Web Tokens(JWT). JWT una herramienta que tiene varias funciones, la principal es que la información del usuario la tiene el usuario mismo en el navegador, de forma de que esta no pueda ser vulnerada. Es una medida de seguridad. A partir de estos, se puede propagar información de forma segura y efectiva, que, además, es verificada, ya que se firma de forma virtual.

La forma en es utilizada la herramienta es que con la info del usuario, a través del hashing que nos provee JWT, se genera un token. Ese token se le da como una cookie al usuario, y de esta forma se le envía esta misma cookie, con el token generado anteriormente, en cada request y de esta manera el servidor sabe que se trata del mismo usuario.

Una aplicación interesante que aprovechamos fue el borrado de cookies, para que luego, si estas quedaban activas, no pudieran ser utilizadas para ir a cualquier parte del sistema sin estar logeado oficialmente, y así corromperlo. De igual forma que si un usuario de tipo administrador se logueó en algún momento, quedando una cookie con su información, y posteriormente un usuario normal se loguea, la idea es que este no pueda acceder al modo administrador de ninguna manera.

4.3. Usuarios

4.3.1. Usuario común

A continuación se muestra la vista que tiene un usuario al entrar al sistema. Como se puede ver, tiene varias pestañas en la parte superior, como el mercado, la billetera, las reglas, las transacciones, las variables, una solapa específica para agregar variables y un log out para terminar la sesión.

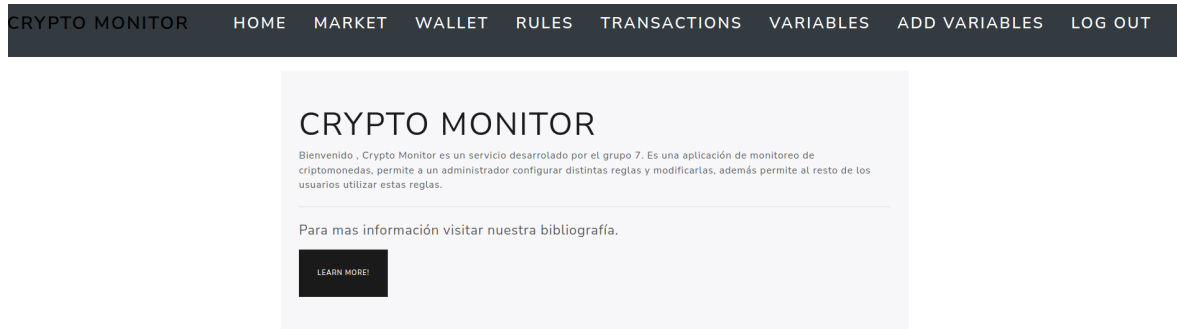


Figura 4.3 – Vista de la pagina principal de usuario.

Al ir a la sección de agregar variable, se puede observar el método para llevar a cabo esta operación.

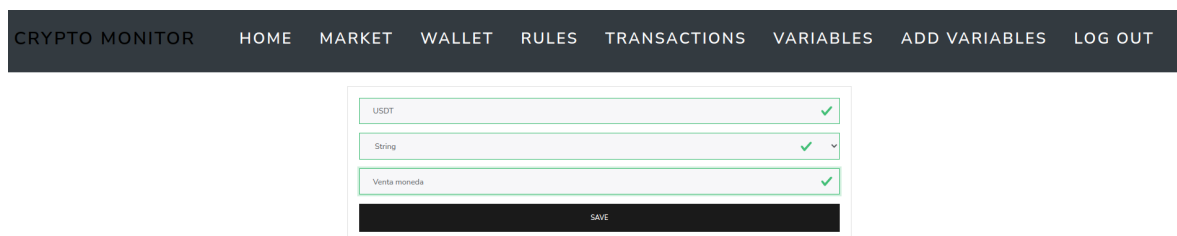


Figura 4.4 – Usuario: agregar variable.

4.3.2. Administrador

Para el modo de administrador, las pestañas que se encuentran en la parte superior son casi las mismas. La diferencia se encuentra en que en este caso no nos encontramos con una que permita agregar variables, pero si una que admite que se agreguen reglas.



Figura 4.5 – Vista del usuario de tipo Administrador.

A modo de ejemplo, se muestra como se agrega una regla, insertando en este caso un archivo json.

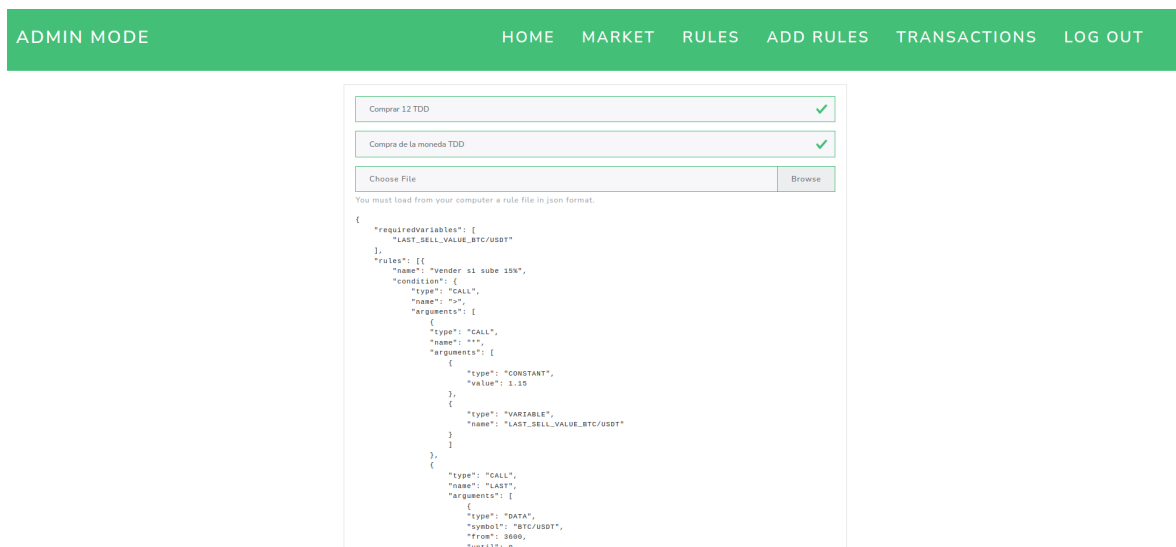


Figura 4.6 – Administrador: agregar regla.

4.4. Binance

Se seteo un entorno de prueba para utilizar valores dados por el servicio externo de Binance, un websocket, obteniendo una billetera con monedas y valores. Para esto se generaron a través de tener asociado a una cuenta de Github un valor de identificación así como también un valor secreto para que se pueda generar una signature junto a un timestamp a la hora de pegarle a los endpoints de Binance y validar nuestra identidad.

Se generó además una conexión a un WebSocket de este servicio externo

5. Conclusión

A modo de conclusión, podemos decir que el presente trabajo nos permitió poner en práctica varios de los conceptos aprendidos durante el desarrollo de la cursada. A su vez, nos hizo familiarizarnos con diversas tecnologías y herramientas muy utilizadas hoy en día, tales como Node, Express y Docker, entre otras, lo cual será muy provechoso para la vida profesional de todos los integrantes de este grupo. Por otro lado, también aprendimos a desarrollar un front-end con login y persistir nuestros datos utilizando una base de datos, en nuestro caso, MongoDB. Por esta razón, consideramos que fue un trabajo muy completo y provechoso, que nos permitió enfrentarnos a problemas reales que surgen en el desarrollo de un proyecto y cómo resolverlos de manera eficaz y en grupo.