Universidad de Zaragoza

Escuela Politécnica de Teruel

Reparto de tareas del proyecto de Sistemas y Tecnologías Web

Escuela Universitaria Politécnica - Teruel

Universidad Zaragoza

Autores:
Alberto Pérez Blasco. 779212
Fernando Revilla Maqueda. 782891
Diego Santomé Rocafort. 777183

14 de junio de 2022

1. Reparto

1.1. Fernando Revilla Maqueda

- **Búsqueda de las APIs.** Tras realizar una búsqueda profunda en internet, se ha decidido trabajar con dos APIs fiables y que presentan datos reales y correctos. Estas dos APIs son:
  - `https://api.coingecko.com`
  - `https://api.preciodelaluz.org`
- **Creación de la BD y Pool de conexiones**. Cuando se obtienen los datos de las APIs, se precisa guardarlos en la base de datos, para ello se crea desde el IDE Netbeans. En el apartado de nueva conexión de base de datos, se selecciona la opción de añadir un nuevo driver. A continuación, se selecciona el archivo h2.jar de Payara. Se incluirán el usuario (stw) y la contraseña así como el JDBC URL. Finalmente se selecciona la opción "PUBLIC"para el esquema.
  Tras obtener éxito en el proceso anterior, se pueden apreciar las diferentes tablas que forman la base de datos.

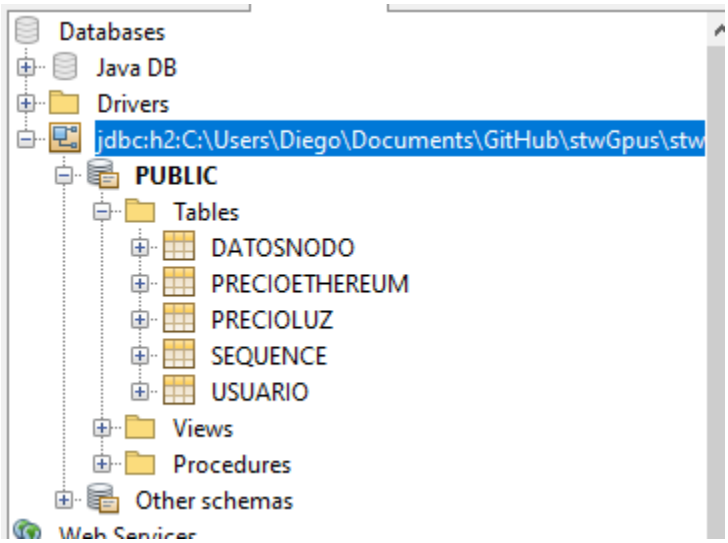

Figura 1: Tablas de la base de datos

- **Almacenamiento de datos en la BD.** Cuando se obtienen los valores en el momento indicado de cada variable, se deben guardar en la base de datos. Desde la clase TimerGetData, se crean los objetos que se quieren almacenar y mediante el objeto de la clase "DAO"pertinente se añade el objeto.

```
public void addEthereumPrice() throws IOException, InterruptedException {
    PrecioEthereum p = new PrecioEthereum();
    p.setFecha();
    p.setPrecio(obtenerPrecioEthereumAPI());
    try {
        precioEthDB.create(p);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Figura 2: Ejemplo añadir en la base de datos un precio de Ethereum

1.2. Alberto Pérez Blasco

- **Creación de API REST.** Para la creación de la API rest, se utilizó la librería Flask de Python, que sirve para crear APIs REST de una forma muy sencilla y con pocas líneas de código. Tras realizar pruebas y commprobar que funcionaba correctamente, se realizó el proceso de obtención de todos los datos obtenidos (enchufe, termómetro y nodo), para unificarlos y de esta forma crear una API REST en la que se podían solicitar los datos
- **Obtención de datos del nodo.** Para obtener los datos del nodo, en primera instancia se realizó un proceso de búsqueda, para saber si había alguna posibilidad de utilizar el propio programa como API REST, debido a que si no, habría que realizar web scraping de la web en local que tiene el programa. Finalmente, para publicar la API REST, hay que añadir unas configuraciones al .bat que lanza el servicio, para que de esta forma publique la API. Los parámetros que hay que añadir, aparecen en el repositorio de GitHub de T-Rex Miner.
- **Obtención de datos de la BD para mostrar en los gráficos.** Para obtener los datos de la BD y mostrarlos en los gráficos, se implementaron unos métodos con los que se obtenían unos objetos específicos dependiendo del gráfico a mostrar. Para obtener los datos de la BD, se utilizan consultas SQL filtrando los datos por la fecha escogida por el usuario, y finalmente, se parsean estos datos para mostrarlos correctamente en los gráficos, siendo este parseo específico para cada uno (limpieza de caracteres, cambio de tipos...)

1.3. Diego Santomé Rocafort

- **Usuarios y sesiones.** Para la la gestión de los usuarios se crearon dos clases para gestionar la tabla usuarios de la base de datos. Cada uno de estos tiene un nombre, apellido, nombre de usuario y contraseña. Los usuarios que entran a la página principal del proyecto tienen la posibilidad de acceder con su usuario creado o crear uno nuevo mediante un servlet que recoge los datos del nuevo usuario.
- **Obtención de datos del termómetro.** Para obtener los datos del termómetro el enchufe se subscribe al topic que está en la Raspberry para que este recoja los datos mediante un ejecu-

table Python que necesita permisos de super usuario que cada vez que se solicita información sobre la tarjeta gráfica también se realiza una lectura de la temperatura.

- **Obtención de datos del enchufe** Para obtener los datos del termómetro el enchufe se subscribe al topic que está en la Raspberry para que este recoja los datos mediante un ejecutable Python. este ejecutable detecta cada mensaje que recibe y lo parsea para almecenar el consumo en un fichero local. 0

1.4. Conjuntamente

Para finalizar, el apartado conjunto que se desarrolló, fue la integración de los gráficos en la página web con la realización del Websocket y los retoques estéticos finales de la página web.