UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

VALPARAÍSO – CHILE

DESARROLLO Y VALIDACIÓN DE UN SISTEMA DE TRAZABILIDAD DE ENERGÍA ELÉCTRICA ENTRE PLANTA FOTOVOLTAICA SALVADOR Y CLIENTE FINAL EN SANTIAGO MEDIANTE SMART CONTRACT BASADO EN TECNOLOGÍA BLOCKCHAIN

DIEGO ALONSO HERMOSILLA ASTORGA

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE

INGENIERO ELECTRICISTA

PROFESORES GUÍAS:

ESTEBAN M. GIL SAGÁS - EDUARDO A. SOTO SEPÚLVEDA

JULIO 2019

En este espacio quisiera agradecer a todas las personas que formaron parte de mi paso por la Universidad, dentro y fuera de ella. A quienes compartieron parte de su vida conmigo en espacios superfluos y también más íntimos. Gracias a todas y todos quienes creyeron en mí, aun cuando el camino ha sido largo y con dificultades, pero también con muchos éxitos y satisfacciones.

Agradecer, también, al fondo basal "FB008" del Centro Avanzado de Ingeniería Eléctrica y Electrónica (AC3E) y a todos los profesores que se esmeraron por entregarme una formación rigurosa a lo largo de estos años, en especial para aquellos que lograron inspirarme y ayudarme a discriminar el camino más coherente a mis principios.

Gracias al equipo de Phineal, por tener la claridad y la motivación para abordar temas de innovación con energías limpias y por apostar con las tecnologías del futuro, por contagiarme de su ímpetu para mirar las cosas de forma diferente y, sobre todo, por su calidez y simpatía.

Gracias a la Rama Estudiantil IEEE USM por compartir "la pera" conmigo y dar espacio a los mini proyectos que siempre cuesta hacer con la carga académica sansana.

Gracias a Wally, Llama, Nacho, Juanito, Lazzy, Enzo, Diego BL, Potro, Pelado, Simón, Jarry, Pajamín y a tantos otros compañeros y compañeras con los que compartí, me reí, fracasé y triunfé.

Gracias a la Ami, que hizo lo imposible maullando, rasguñándome y acostándose en mi teclado para que no terminara esta Memoria.

Por último, gracias a mi familia y a Lilian, por estar ahí siempre.

Resumen

Esta Memoria de Título asume el desafío de validar y fortalecer el sistema de trazabilidad de energía eléctrica de la empresa *Phineal* a través de la programación de un *smart contract* desplegado en *Ethereum* que, gracias a una *aplicación descentralizada* (*DApp*), facilita la reproducción de una *cadena de suministro* instantánea de electricidad basada en *Blockchain GTIME* (*blockchain federada* desarrollada por *Phineal*). En el problema de trazabilidad, la planta PV Salvador es el punto de generación, la oficina de *Phineal* el punto de retiro y un modelo simplificado del Sistema Eléctrico Nacional (SEN) es el vínculo que permite incorporar las pérdidas por transporte entre ambos agentes. Usando el modelo del SEN, y de acuerdo con el marco regulatorio actual en Chile, se decidió emplear el Método de Bialek con Flujos Gruesos (a nivel de transmisión nacional) y el Método de Estampillado Postal (a nivel de transmisión zonal y distribución) para desplazar las pérdidas hacia los clientes finales. A fin de conseguir una *cadena de suministro* instantánea y autoejecutable, dichos procedimientos se incluyeron en un servidor ficticio del Coordinador Eléctrico Nacional (base de datos simulada) y en un archivo JavaScript de la propia *DApp*, respectivamente. La *DApp*, mediante el *smart contract*, utiliza una herramienta descentralizada de lectura web llamada *Oraclize* para importar los datos de entrada, evitando que esta se transforme en un intermediario en el que haya que depositar confianza. De este modo, los datos solicitados por el *smart contract* hacia los servidores implicados en el sistema de trazabilidad son:

- Inyección de la planta y retiro del cliente final; medidos por los nodos *phiNet* y *phiNergy* de *Phineal*. Datos extraídos desde el *Server* de *Blockchain GTIME* (unidad de almacenamiento).

- Factor de proporción de generación horaria de PV Salvador, *demanda gruesa* horaria y *demanda real* horaria; calculados a partir de la aplicación del Método de Bialek sobre el modelo simplificado del SEN en 14 estados horarios definidos para efectos de simulación. Datos extraídos desde un servidor ficticio del Coordinador (dispuesto para este prototipo).

Luego, en conocimiento de toda la información de entrada, la *DApp* estima las pérdidas eléctricas empleando el Método de Estampillado Postal y construye una cadena de suministro de las cantidades físicas involucradas (en $[MWh]$ y $[kWh]$). Finalmente, para efectos de este prototipo, el contacto con el usuario se logra mediante la representación gráfica de la trazabilidad de flujos proyectada en la dirección de "loopback" disponible en cualquier navegador web; permitiendo al cliente final estar al tanto del abastecimiento energético a tiempo real. De este modo es posible conseguir, para un determinado instante de tiempo y en rangos de 20 minutos, una *cadena de suministro* instantánea que cuenta con dos niveles de validación en sus datos de origen (*Blockchain GTIME* y *Ethereum*), para su uso en certificados de producción, de abastecimiento y de *cadena de suministro* de "electricidad verde".

Abstract

This document takes on the challenge of validating and strengthening *Phineal*'s electricity traceability system through the programming of a *smart contract* deployed in *Ethereum* which, thanks to a *decentralized application* (*DApp*), facilitates the reproduction of an instantaneous electricity *supply chain* based on *Blockchain GTIME* (*federated blockchain* developed by *Phineal*). In the traceability problem, the PV Salvador plant is the point of generation, *Phineal*'s office is the point of consumption and a simplified model of the National Electric System (SEN) is the link that allows the incorporation of transport losses between both agents. Using the SEN model, and in accordance with the current regulatory framework in Chile, it was decided to use the Bialek Method with Gross Flows (at the national transmission level) and the Postal Stamping Method (at the zonal transmission and distribution level) to displace the losses to the final customers. In order to achieve on instantaneous and self-executing *supply chain*, these procedures were included in a fictitious server of the National Electrical Coordinator (simulated database) and in a JavaScript file of the *DApp* itself, respectively. The *DApp*, through the *smart contract*, uses a decentralized web-reading tool called *Oraclize* to import the input data, preventing it from becoming a trusted intermediary. In this way, the data requested by the *smart contract* to the servers involved in the traceability system are:

- Plant injection and final customer consumption; measured by *Phineal*'s *phiNet* and *phiNergy* nodes. Data extracted from the *Blockchain GTIME Server* (storage unit).

- Proportion factor of generation per hour of PV Salvador, gross demand per hour and actual demand per hour; calculated from the application of the Bialek Method on the simplified model of the SEN in 14 states defined for simulation effects. Data extracted from a fictitious server of the Coordinator (available for this prototype).

Then, given all input information, the *DApp* estimates the electrical losses using the methodological proposal and constructs a *supply chain* of the physical quantities involved (in $[MWh]$ and $[kWh]$). Finally, for the purposes of this prototype, contact with the user is achieved through the graphical representation of the projected flow traceability through the "loopback" address available in any web browser; allowing the final customer to be aware of the energy supply in real time. In this way it is possible to achieve, for a given instant of time and in ranges of 20 minutes, an instantaneous *supply chain* that has two levels of validation in its data of origin (*Blockchain GTIME* and *Ethereum*), for use in certificates of production, of consumption and *supply chain* of "green electricity".

Índice

1. Introducción **1**

1.1. Trazabilidad de la energía eléctrica: presente y futuro 1

1.2. Definición del problema y objetivos 6

1.3. Alcances del trabajo 7

2. Blockchain y Smart Contracts: definiciones y conceptos **8**

2.1. Tecnología blockchain 8
2.1.1. Definición 8
2.1.2. Funcionamiento 9
2.1.2.1. Descripción básica 9
2.1.2.2. Principales características 11
2.1.2.3. Ciclo de encadenamiento de bloques 14
2.1.3. Taxonomías 15

2.2. Smart contract: una poderosa aplicación de blockchain 17
2.2.1. Definición 17
2.2.2. Funcionamiento 17
2.2.2.1. Descripción básica 17
2.2.2.2. Principales características 18
2.2.2.3. Interacción con smart contract 18

2.3. Aproximación de blockchain al sector energético 20

3. Sistema de trazabilidad de energía basado en tecnología blockchain **25**

3.1. Blockchain GTIME: el blockchain de energía 25

3.2. Prototipo tecnológico del sistema de trazabilidad 29
3.2.1. Taxonomía blockchain 29
3.2.2. Componentes básicos y funcionamiento del sistema GTIME 30
3.2.2.1. Medidores de generación/demanda 32
3.2.2.1.1. phiNergy 32
3.2.2.1.2. Red de estaciones phiNet 34
3.2.2.2. Miners 35
3.2.2.3. Server y Transaction 38

4. Métodos de asignación de pérdidas en transmisión como herramientas para la trazabilidad de la energía eléctrica **41**

4.1. Método de Bialek 44
4.1.1. Algoritmo de inyecciones de potencia 45
4.1.2. Algoritmo de retiros de potencia 47
4.1.3. Rastreo de electricidad según asignación de pérdidas 50

4.2. Método de Prorrateo o Estampillado Postal 51

4.3. Cuadro comparativo de las metodologías estudiadas 53

5. Aplicación descentralizada, programación de Smart Contract y validación del sistema de trazabilidad de energía eléctrica bajo el protocolo GTIME **54**

5.1. Validación de trazabilidad de flujos entre producción y consumo 54
5.1.1. Modelo simplificado del Sistema Eléctrico Nacional 56
5.1.2. Aplicación de Flujos Gruesos de Bialek en modelo simplificado del SEN sujeto al estado de la red del 28 de febrero de 2018 a las 14:00 hrs. 62
5.1.2.1. Resultados 63
5.1.2.2. Análisis general 65
5.1.2.3. Construcción de cadena de suministro a nivel nacional 67
5.1.3. Aplicación de Estampillado Postal en modelo simplificado del SEN sujeto al estado de la red del 28 de febrero de 2018 a las 14:00 hrs. 69
5.1.3.1. Resultados 70
5.1.3.2. Análisis general 70
5.1.3.3. Construcción de cadena de suministro a nivel zonal y de distribución 71
5.1.4. Resumen previo a la implementación 74

5.2. Desarrollo de Aplicación Descentralizada y programación de Smart Contract 75
5.2.1. Descripción general de la DApp 76
5.2.2. Smart Contract - EnergyTraceability.sol 79
5.2.3. Archivo JavaScript - App.js 82
5.2.4. Ejecución de la DApp 87

6. Conclusiones **90**

7. Referencias **92**

8. Anexos **97**

8.0 Resumen Ejecutivo 97

8.1. Anexos Capítulo 1 104

8.2. Anexos Capítulo 2 104
8.2.1. Minería de datos, protocolo de consenso y bifurcaciones en la red 104
8.2.2. Puntos fuertes y débiles de blockchain 115
8.2.2.1. Ventajas 115
8.2.2.2. Desventajas 116

8.3. Anexos Capítulo 3 118
8.3.1. Información técnica del prototipo tecnológico de Blockchain GTIME 118
8.3.2. Minería de datos, protocolo de consenso y escalabilidad del historial 123
8.3.3. Aplicaciones: Sello Sol, el blockchain solar de Latinoamérica 126

8.4. Anexos Capítulo 4 129
8.4.1. Ejemplo del Método de Bialek usando Flujos Gruesos 129
8.4.2. Código MATLAB Ej. Sistema de Garver - M. de Bialek Flujos Gruesos 135
8.4.3. Ejemplo del Método de Bialek usando Flujos Medios 139
8.4.4. Código MATLAB Ej. Sistema de Garver - M. de Bialek Flujos Medios 144
8.4.5. Ejemplo del Método de Bialek usando Flujos Netos 148
8.4.6. Código MATLAB Ej. Sistema de Garver - M. de Bialek Flujos Netos 152
8.4.7. Método de Kirschen, Allan y Strbac 156
8.4.7.1. Conceptos fundamentales 157
8.4.7.1.1. Dominio de un generador 157
8.4.7.1.2. Área común de generación 158
8.4.7.1.3. Enlaces del sistema 159
8.4.7.2. Representación gráfica del sistema 159
8.4.7.3. Método recursivo del sistema 159
8.4.8. Ejemplo del Método de Kirschen, Allan y Strbac 161
8.4.9. Código MATLAB Ej. SEP de prueba - M. de Kirschen, Allan y Strbac 168
8.4.10. Ejemplo Método de Prorrateo o Estampillado Postal 171

8.5. Anexos Capítulo 5 174
8.5.1. Datos de entrada para la aplicación de Flujos Gruesos de Bialek en modelo simplificado del SEN sujeto al estado de la red del 28 de febrero de 2018 a las 14:00 hrs. 174
8.5.2. Resultados Flujos Gruesos de Bialek en modelo simplificado del SEN sujeto al estado de la red del 28 de febrero de 2018 a las 14:00 hrs. 176
8.5.3. Código MATLAB en SEN reducido - Método de Bialek Flujos Gruesos 178
8.5.4. Esquema DigSilent PF del SEN reducido con flujo de carga AC sujeto al estado de la red del 28 de febrero de 2018 a las 14:00 hrs. 182

1. Introducción

1.1 Trazabilidad de la energía eléctrica: presente y futuro

Según un artículo publicado por la Asociación de la Industria Eléctrica y Electrónica de Chile (AIE) [28], la trazabilidad de un producto o servicio puede entenderse como el conjunto de acciones y métodos que permiten registrar e identificar cada actividad por la que atraviesa este, desde su génesis hasta que llega al cliente final. El seguimiento se obtiene gracias al uso de herramientas idóneas para dar con el historial, la ubicación y la trayectoria del producto a lo largo de toda su *cadena de suministro*.

De la definición anterior se desprende que la energía eléctrica, entendida como producto, también admite el rastreo de las etapas que facilitan su abastecimiento, como lo son: generación, transmisión y distribución. Para ello, se emplean instrumentos de telemetría y sistemas de inscripción y almacenamiento de datos, que son particularmente útiles tanto en las labores de facturación como de continuidad en la cadena de pagos del sistema [38].

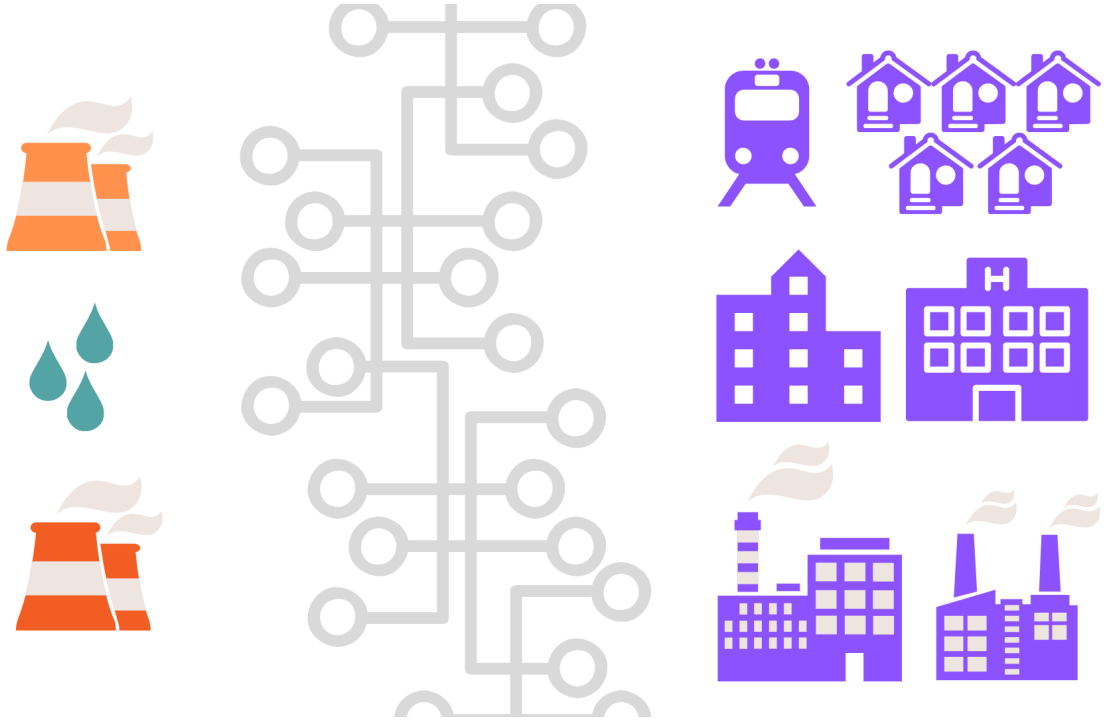


Figura 1.1: El sistema tradicional centraliza la producción y el consumo de electricidad. Elaboración propia.

Con el esquema tradicional que sugiere la *Figura 1.1*, en donde la producción y cesión de energía eléctrica se articula para abastecer a clientes finales que pagan por un servicio en el que no participan,

se persigue ilustrar que la *cadena de suministro* de electricidad tiende a asignar a cada agente del sistema eléctrico de potencia (SEP) un rol claramente definido, siendo la trazabilidad tan sólo un mecanismo de comprobación de las cantidades pactadas en contratos unidireccionales en lo que a potencia, energía y dinero refiere (ver *Figura 1.2*). Además, dado que cualquier movimiento de flujo sobre el SEP está sujeto a la *ley de conservación de la energía*, resulta natural que la *cadena de suministro* incorpore también las pérdidas por *Efecto Joule* cometidas durante el transporte. Estas, que por su carácter no lineal representan un desafío en el rastreo y determinación de los agentes que las producen, se asignan en general mediante factores de cobro de diversa índole (como la determinación del precio de nudo promedio del sistema de transmisión nacional y zonal, por ejemplo [39]) dejando de lado el aspecto físico del problema [37][40].

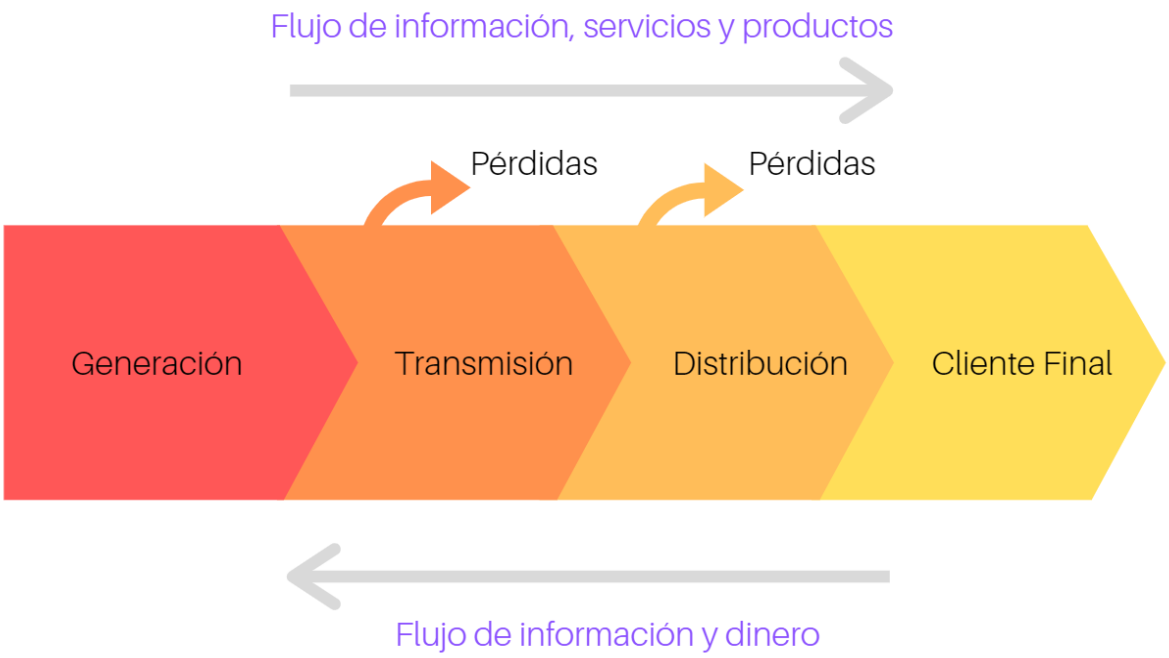


Figura 1.2: Cadena de suministro de electricidad en formato tradicional. Los generadores abastecen la demanda lejos de los centros de consumo y los clientes finales adoptan las pérdidas eléctricas cometidas durante la etapa de transporte. Elaboración propia.

No obstante, ante la penetración de las nuevas tecnologías de generación basadas en energías renovables y de aquellas asociadas al control de variables eléctricas vía *Internet de las Cosas* (*IoT*, por sus siglas en inglés), así como de las exigencias ciudadanas por mayor transparencia en los procesos productivos y ambientales, el sector Energía se ha visto forzado a repensar el concepto de trazabilidad para fortalecer la *cadena de suministro* de la "electricidad verde", debido a que esta puede representar un elemento diferenciador importante a nivel comercial en la medida en que los

consumidores orienten sus preferencias hacia productos o servicios con baja huella de carbono y sean capaces de generar su propia energía. Lo anterior, además de potenciar la *cadena de valor* de la energía proveniente de fuentes renovables, implica que ya no bastaría con conocer la naturaleza del parque generador o de la demanda, sino que, conforme los consumidores se vayan transformando en ciudadanos productores o "prosumidores", la demanda tomará un rol más activo en el mercado, pudiendo escoger a su proveedor energético según los precios de compra/venta, del impacto ambiental de la tecnología, de las externalidades positivas del intercambio *peer to peer*, e inclusive, del carácter político de la generación. Esto último adquiere sentido pues, bajo algunas circunstancias, un modelo energético descentralizado puede funcionar en armonía con el concepto de economía colaborativa [29][34].

En relación a la "actualización" del nuevo concepto de trazabilidad, F. Orecchini, A. Santiangeli, *et al.* [27] toman como ejemplo el caso de los vehículos eléctricos y las futuras estaciones de servicio, donde el conocimiento de origen de las fuentes de energía utilizadas para cargar la batería de un auto eléctrico funcionaría como un certificado que muestra la cantidad de emisiones de gases de efecto invernadero ahorradas a la atmósfera, dándole la oportunidad al consumidor final de discriminar cuáles son las estaciones de servicio que ofrecen recargas "verdaderamente limpias". Por lo que, como explican los autores, mientras el conjunto de instalaciones destinado a electromovilidad esté vinculado a *redes inteligentes* y a potenciar el uso de renovables, el énfasis de la trazabilidad estaría puesto en determinar el recurso energético primario, la ubicación de la planta generadora, una variante de identificación temporal, y obviamente, las cantidades de energía inyectadas/retiradas en el sistema.

En ese sentido, la trazabilidad de la producción eléctrica mediante *blockchain* representa una oportunidad para materializar lo discutido en la conferencia [27], puesto que, en función de los conceptos que se abordarán en el Capítulo 2, facilita el monitoreo y almacenamiento seguro de toda información relativa a la *cadena de suministro* de la energía. Con esta tecnología, el seguimiento de las transferencias energéticas se hace sin que sea necesario depositar confianza en una institución validadora o en auditorías externas, puesto que toda medición, registro y procesamiento de datos se hace, simultáneamente, en dispositivos electrónicos distribuidos que replican la información sin intervención humana, evitando la centralización y vulnerabilidad de esta.

Lo anterior derivaría en la posibilidad de que los agentes de un SEP interactúen mediante contratos que se autoejecutan de manera descentralizada y en lenguaje máquina (*smart contracts*), con los cuales será posible efectuar el pago correspondiente por uso del sistema de transporte, rastrear el origen de la electricidad consumida, asimilar las pérdidas incurridas en transmisión y/o distribución,

estar al tanto de las emisiones de carbono que involucra el abastecimiento y, en definitiva, tener acceso a cualquier otra información relevante que persiga agregar valor al producto o servicio para el que se requiere dicha energía.

Al respecto, el esquema de la *Figura 1.3* persigue representar la interacción entre diferentes actores a través del ícono de conexión inalámbrica (). En la figura se pretende enseñar la posibilidad de que grandes generadoras renovables puedan establecer contratos de suministro con empresas de producción y de transmisión, pero también lo podrán hacer pequeños generadores distribuidos con empresas que busquen producción limpia, empresas que ofrezcan puntos de carga para vehículos eléctricos, instituciones que deseen certificar sus inyecciones o retiros, o bien, para el comercio en pequeñas comunidades energéticas, entre otras. Bajo cualquier situación, los agentes tendrán acceso al historial de movimientos de activo a través de plataformas web, pudiendo comprobar el origen y las cantidades transadas en cada contrato de suministro (piénsese en energía o valorización de esta).

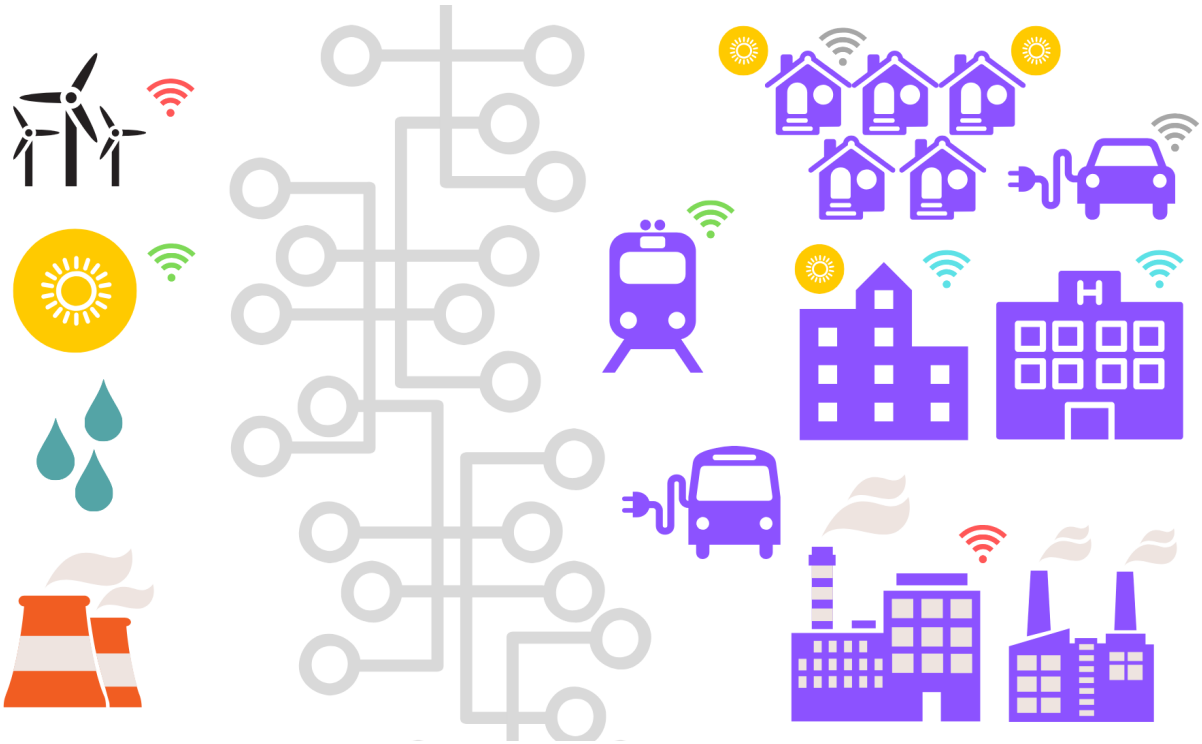


Figura 1.3: La descentralización de la producción energética promueve la interacción entre agentes de distinto tamaño, diversificando la matriz energética y potenciando la participación de los prosumidores. Elaboración propia.

En lo que concierne a la interacción de activos digitales vinculados a la energía, en el artículo *Blockchain y energía: la revolución de los prosumidores* publicado en el blog de la empresa *Phineal* [23], se comenta que la *tokenización* de las unidades de energía transada podría abrir un mercado de

intercambio de activos seguro y transparente en el que cada *token* sirva como medio de pago, modificando por completo la *cadena de suministro* tradicional (ver *Figura 1.4*).

Esta idea se refuerza por los autores del artículo [16], quienes sostienen que los *tokens* no se deben vincular necesariamente a una criptomoneda, ya que estos pueden usarse como verdaderos "certificados" que garantizan la ocurrencia de eventos en determinados instantes de tiempo. Como, por ejemplo, la inyección de $1[kWh]$ proveniente desde una instalación fotovoltaica y su posterior retiro, facilitando la trazabilidad de los activos.

En consecuencia, como sugiere la *Figura 1.4*, la idea del consumidor pasivo habrá quedado atrás, dando paso a un prosumidor cada vez más involucrado en el manejo de su energía, tanto para fines propios como para fines colectivos. Así, la descentralización de la producción energética dará paso a un bloque transmisor fusionado por el que circulen inyecciones y retiros de forma bidireccional, y cuyo volumen de información, para efectos de trazabilidad como para valorizar las transferencias efectuadas, deberá ser gestionado por mecanismos de medición, registro y cómputo más transparentes, basados en *IoT*, mediante *aprendizaje máquina*, o en términos amplios, con *inteligencia artificial*.

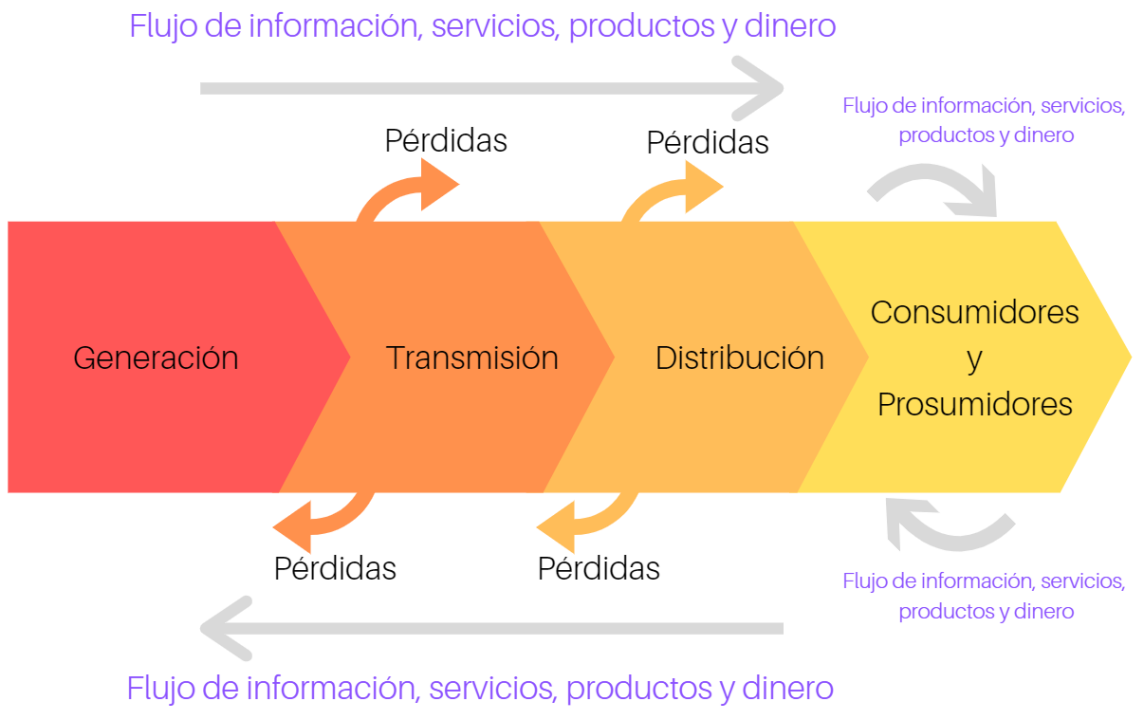


Figura 1.4: Futura cadena de suministro. La irrupción de prosumidores fomentará la venta de energía más cerca de los pequeños consumidores. Los clientes finales podrán escoger al *pool* de generadores más baratos y limpios. El bloque transmisor se fusionará y robustecerá para la bidireccionalidad de flujos, debiendo ser administrado por tecnologías más inteligentes. Elaboración propia.

1.2 Definición del problema y objetivos

Desde la óptica de la ingeniería eléctrica, la trazabilidad de la energía puede entenderse como el rastreo o seguimiento instantáneo de los flujos de potencia inyectados a la red. Lo anterior permitiría responder algunas cuestiones propias de la trazabilidad eléctrica como: el origen o generación, el desplazamiento de flujos a través de las líneas y el punto de recepción o consumo. En virtud de ello, uno de los factores más relevantes es la transmisión, debido a que esta permite la interacción entre los productores y consumidores del mercado eléctrico. De este modo, si se conoce el nivel de uso que cada agente del SEP hace del conjunto de líneas e instalaciones de transmisión, se puede establecer un cargo físico por la distribución de flujos, y si se está en conocimiento de las cantidades inyectadas y retiradas del sistema, es posible completar la *cadena de suministro* de electricidad. Con lo anterior, se subentiende que una consecuencia ineludible del problema de trazabilidad son las pérdidas eléctricas vinculadas al *Efecto Joule*. Al respecto, dado que no es posible rastrear su procedencia debido a la dependencia no lineal de la corriente circulante por las líneas, en este trabajo se presentan algunos métodos que asumen el desafío de cuantificar las pérdidas totales para asignarlas a los agentes del SEP, en virtud de la contribución que estos tienen sobre las transferencias de potencia del sistema.

De este modo, en sintonía al título de esta Memoria, en el presente documento se pretende mostrar los conceptos clave para desarrollar un *smart contract* que termine por validar un sistema de trazabilidad de energía eléctrica con *blockchain*. En términos concretos, se realizará una *aplicación descentralizada* que, adoptando una propuesta metodológica de asignación de pérdidas, permita construir, de manera instantánea en períodos de 20 minutos, una *cadena de suministro* de electricidad entre la planta PV Salvador y un cliente final en Santiago. Para incorporar los datos empíricos de inyección y retiro se utilizará la plataforma de energía *Blockchain GTIME* diseñada por la empresa *Phineal* [26], mientras que, para simular la obtención instantánea de las pérdidas incurridas entre ambos puntos a lo largo de la red de transporte, se tomará como referencia un modelo simplificado del sistema de transmisión nacional en 14 estados horarios construidos con información empírica del portal de información pública del Coordinador Eléctrico Nacional [38], excluyendo el cálculo de las pérdidas a nivel de transmisión zonal y de distribución.

Formalmente, los objetivos del trabajo son los siguientes:

Objetivo general:

- Desarrollar una propuesta metodológica que permita validar, a través de un *smart contract* basado en tecnología *blockchain*, un sistema de trazabilidad de energía eléctrica entre la Planta Fotovoltaica Salvador y un cliente final en la ciudad de Santiago con el propósito de certificar

que la electricidad consumida ha sido generada total o parcialmente utilizando el recurso solar como fuente energética primaria.

Objetivos específicos:

- Desarrollar una propuesta metodológica que permita representar la transacción energética entre la planta fotovoltaica y el cliente final mediante el uso de modelos de transferencia de potencia.

- Programar, vía software, un *smart contract* que integre las restricciones de la metodología planteada, el concepto *GTIME* elaborado por *Phineal* y la dinámica de precios involucrada en la transacción energética; con objeto de definir las variables y condiciones necesarias para asegurar que la electricidad consumida ha sido generada total o parcialmente con energía solar.

- Implementar el *smart contract* sobre un prototipo tecnológico del sistema de trazabilidad de energía con el fin de validar, en todo momento, la transacción de datos entre generación y consumo por medio de una plataforma tipo *blockchain*.

1.3 Alcances del trabajo

Este documento se limita a la obtención de una *cadena de suministro* instantánea de electricidad a nivel físico, recurriendo a equipos de medición y a simulaciones para conseguir las cuantías pertinentes. Para lograr los objetivos de esta Memoria se dispuso del estudio de cuatro métodos de asignación de pérdidas en transmisión que trabajan con flujos totales (no incrementales), de los cuales tres se redactaron formalmente y sólo dos se emplearon en la propuesta metodológica, sin considerar el cálculo de pérdidas a nivel zonal ni en distribución. Con el fin de construir empíricamente la *cadena de suministro* instantánea entre la planta PV Salvador y el cliente final en Santiago, se utilizaron las mediciones de inyección y retiro almacenadas en la *blockchain federada* de *Phineal*, mientras que para despejar las pérdidas se usaron los registros de generación y demanda reales publicados por el Coordinador para un día arbitrario del año 2018 y sobre un modelo simplificado del Sistema Eléctrico Nacional. En relación con las herramientas tecnológicas involucradas, es bueno comentar que se estudiaron diversos artículos, papers, revistas y blogs web para abordar los conceptos *blockchain* y *smart contract*. También, se trabajó en conjunto con la empresa *Phineal* para tener acceso a toda la información técnica y conceptual de *Blockchain GTIME* a fin de desarrollar la *aplicación descentralizada*. En ella, se incluye un *smart contract* desarrollado en lenguaje Solidity sobre una red de prueba de *Ethereum* y un archivo *JavaScript* extraído desde una web colaborativa, el cual se adaptó a las necesidades del sistema de trazabilidad de energía perseguido en este documento.

2. Blockchain y Smart Contracts: definiciones y conceptos

Con el fin de adentrar al lector a los fundamentos tecnológicos de esta Memoria, este capítulo mostrará las definiciones y conceptos más importantes para comprender qué es *blockchain* y qué es un *smart contract*; siendo la antesala para contextualizar la plataforma desarrollada por *Phineal* que se encarga de efectuar la trazabilidad de la producción energética de plantas generadoras.

2.1 Tecnología blockchain

La tecnología *blockchain* o "*cadena de bloques*" fue introducida por primera vez en un artículo técnico publicado el año 2008 bajo la autoría de Satoshi Nakamoto con el propósito de eliminar el problema de "doble gasto" (double-spending) relativo a las transacciones comerciales vía internet [5]. Si bien el legado más visible del trabajo de Nakamoto fue la materialización de *Bitcoin*, lo esencialmente disruptivo no fue la creación de este nuevo medio de pago, sino la plataforma online sobre la cual se sustentan las operaciones de dicha criptomoneda: la *blockchain*.

En el contexto de *Bitcoin*, la *blockchain* es la plataforma que da vida al intercambio de dinero digital de forma directa o *peer to peer* entre emisor y receptor [5]. No obstante, el concepto *blockchain* es mucho más universal que el comercio con criptomonedas *bitcoins* (BTC), ya que esta permite la transferencia de *activos digitales* de cualquier tipo, siendo el intercambio de dinero sólo una de las tantas aplicaciones que se pueden desarrollar sobre la tecnología de bloques.

En lo que concierne a este capítulo, se pretende describir *blockchain* basándose en el material bibliográfico estudiado, con un claro énfasis en los conceptos, procesos e ideas que serán útiles para entender el sistema de trazabilidad de energía eléctrica desarrollado por *Phineal*.

2.1.1 Definición

A pesar de que aún no existe una definición formal para *blockchain*, diversas personas, empresas e instituciones de los círculos de innovación tecnológica y del sector financiero se han aventurado a conceptualizar esta nueva tecnología. Considerando las definiciones propuestas en el material bibliográfico de este trabajo (a saber: [5], [8], [10], [11] y [15], por citar algunos), es posible afirmar que la tecnología *blockchain* es un nuevo tipo de "base de datos"; se trata de un registro inmutable de transferencias de *activo digital* organizado en bloques secuenciales de información, validado, sostenido y almacenado por una red distribuida de computadores.

El hecho de que la red distribuida carezca de una autoridad de validación central se traduce en la exclusión total de intermediarios en los procesos, facilitando la transparencia y trazabilidad de estos, disminuyendo los costos pertinentes a cada transacción, además de generar una mayor simetría de información en los entornos o mercados en los que esta tecnología se aplique.

2.1.2 Funcionamiento

2.1.2.1 Descripción básica

Si un conjunto de usuarios desea intercambiar un *activo digital* en ausencia de un intermediario de confianza, entonces, hay cabida para una red con *blockchain*. En una red de este tipo, el registro de cada operación no es responsabilidad exclusiva de un servidor central, sino de múltiples nodos que tienen una réplica actualizada de la base de datos (ver *Figura 2.1*).

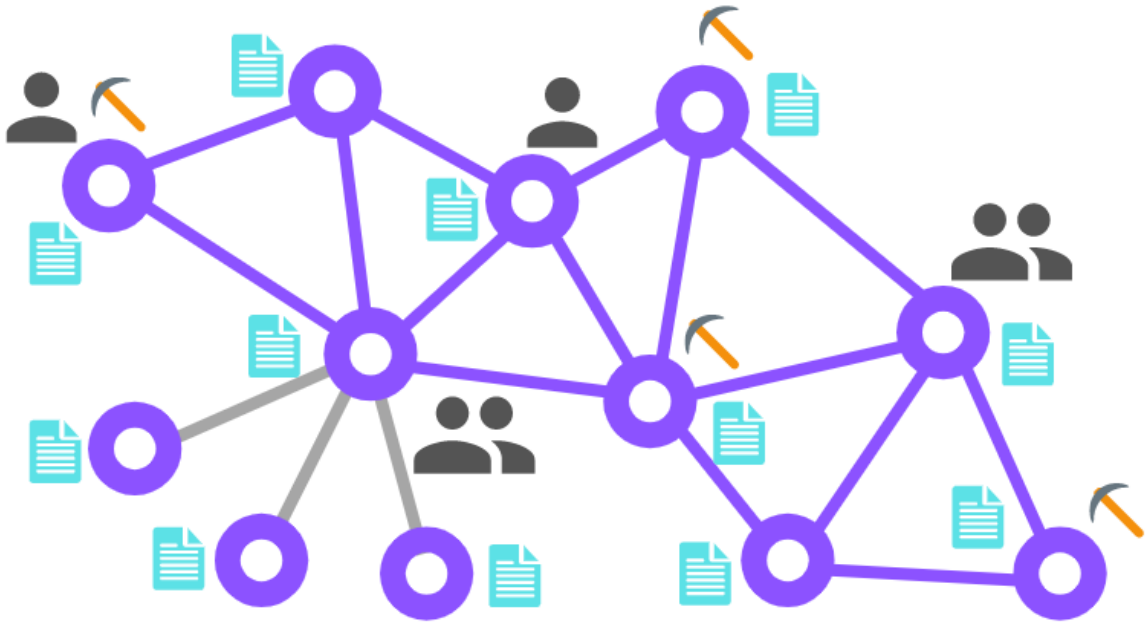


Figura 2.1: Esquema de una red con *blockchain*. La topología de red de información distribuida permite que los nodos se vinculen directamente para efectuar, registrar y almacenar las transacciones en la base de datos; de la cual, todos tienen una copia (la *blockchain* misma, en celeste). Todos los nodos cuentan con una *dirección* dentro de la red y pueden estar compuestos por más de un usuario. Algunos nodos usan la red para efectuar transferencias y otros se encargan de validar las transacciones según el *protocolo de consenso* (nodos *mineros*, simbolizados por una picota). Si bien ningún nodo es más influyente que otro, igualmente se pueden admitir nodos de configuración descentralizada (enlaces en gris), donde se replica la base de datos en los nodos conectados, pero estos no tienen vínculo alguno con el resto de la red. Elaboración propia.

En dicho entorno distribuido (o descentralizado, como mínimo), cuando uno de los usuarios envía un *activo digital* a otro, el resto de los participantes se encarga de comprobar la veracidad de la transacción y, una vez aprobada, esta se guarda criptográficamente en un bloque o fichero, donde se junta con todas las transferencias validadas en el mismo estado del sistema, esto es, durante un período de tiempo de recolección de datos definido previamente en la red.

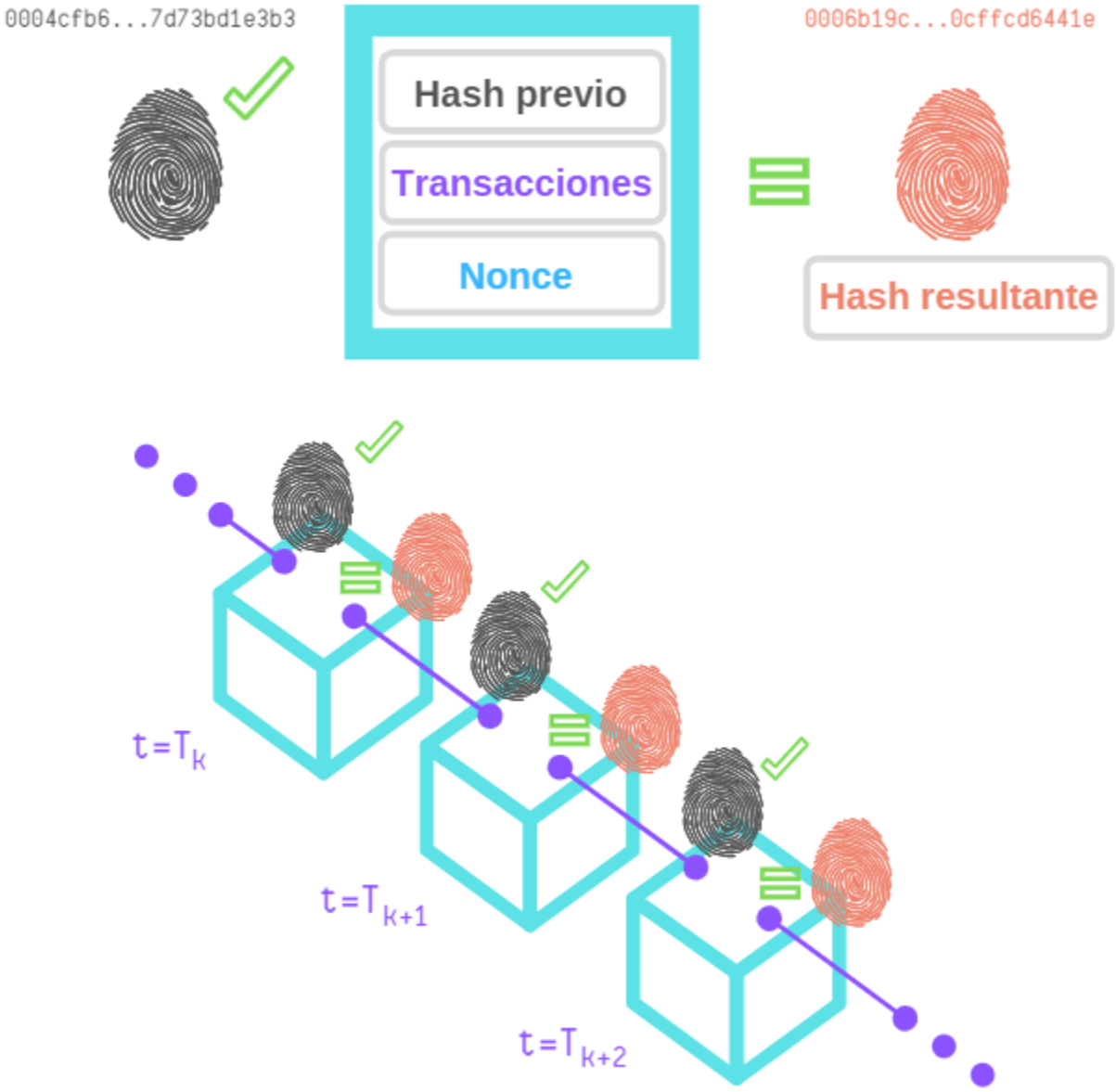


Figura 2.2: Representación del contenido de un bloque y de la secuencia de concatenación de estos en *blockchain*. Cuando la red propone un nuevo bloque, el código *hash* encontrado por alguno de los *mineros* de la red permite "sellar" el bloque candidato, adhiriéndolo a la *cadena* con una *marca de tiempo* y validando las transacciones contenidas dentro de él. El *hash* es único para cada bloque y los *mineros* lo buscan según el *protocolo de consenso* programado en la red. El *hash* encontrado debe ser igual al resultado de una función criptográfica que toma como parámetros de entrada el *hash* del bloque anterior (*hash previo*), las transacciones en cuestión y el número aleatorio *nonce*. Elaboración propia.

Dicho proceso de validación descentralizada permite obtener, en función del contenido del bloque, un código alfanumérico de longitud fija llamado *hash* que hace las veces de "huella digital" del bloque en cuestión (ver *Figura 2.2*). El *hash* es único e irrepetible para cada bloque y su código se hereda secuencialmente, imponiendo un orden y sirviendo de "eslabón" entre los bloques que conforman la *cadena*. Advierta que, debido a que las transacciones dependen del estado del sistema, el orden secuencial de la *blockchain* obedece, también, a una cuestión cronológica. Esto confiere a los usuarios la posibilidad de rastrear la evolución de las transacciones del *activo digital*; llegando, incluso, al bloque que da origen a la *cadena* de datos (*génesis*).

Finalmente, con la adición de un nuevo bloque bajo la aprobación de la comunidad de participantes, la *blockchain* se actualiza y se replica en todos los computadores de la red distribuida. Lo que sigue es la postulación de nuevos bloques con sus respectivas transferencias sujetas a validación, dado que la construcción de la *cadena* resulta de un proceso cíclico o repetitivo.

2.1.2.2 Principales características

En conformidad a las fuentes consultadas y en coherencia al ítem anterior, las características más relevantes y diferenciadoras de la tecnología *blockchain* se pueden resumir en cinco grandes aspectos:

- **Consenso:**
  La validación de las transacciones las hace un conjunto de usuarios a través de sus computadores (nodos) en tiempo real y no una entidad central en plazos impuestos. El *protocolo de consenso* logra que todos los participantes de la red distribuida lleguen a un acuerdo sobre la veracidad del historial de transferencias compartido, evitando que se produzca el "doble gasto" en las operaciones [5] y reemplazando la aprobación central por una validación distribuida. Este *consenso* se lleva a cabo gracias a los nodos validadores de la red denominados *mineros* (ver *Figura 2.1*), quienes, a través de un algoritmo criptográfico programado en sus computadores, consiguen la "huella digital" o *hash* de cada bloque del sistema. Para *Bitcoin*, por ejemplo, el *consenso* se hace mediante la *prueba de trabajo* o *proof of work* y la función criptográfica utilizada es la SHA-256, con la cual se obtiene un *hash* de 64 dígitos hexadecimales de un tamaño fijo de 256 bits [5]. Gracias a esta "huella digital" los usuarios pueden determinar la versión aceptada de la *blockchain* en cualquier momento, consensuándolos respecto de la validez de los datos. Adicionalmente, se debe comentar que, dependiendo de la taxonomía de la red con *blockchain*, la

mantención del *protocolo de consenso* puede tener un costo computacional y de energía elevados para cada *minero* de la red, ya que la obtención de *hashes* puede estar sujeta a incentivos monetarios o recompensas que promueven la competencia por el *minado* de los bloques.

Obs: Para ahondar en los conceptos *minería* y *protocolo de consenso* diríjase al ítem 8.2.1 de la sección Anexos, al final de esta Memoria.

- **Concatenación:**

  La información pertinente a cada transacción se encripta y agrupa en bloques que posteriormente, enlazados vía *hash*, conformarán la *blockchain* (ver *Figura 2.2*). La estructura general de cada bloque o fichero está definida por: el *hash* del bloque anterior (*hash génesis* si se tratara del primer bloque), el contenido criptográfico de las transferencias y un código alfanumérico al azar denominado *nonce*. Este último número es la variable que los *mineros* deberán iterar, bajo las reglas impuestas por el *protocolo de consenso* adoptado en la red distribuida, para encontrar el *hash* del bloque que será acoplado posteriormente.

- **Inmutabilidad:**

  Según Ó. Lage [12], el uso de algoritmos criptográficos en *blockchain* tiene por objetivo proteger la cadena; otorgando autenticación, integridad y no repudio en la red. En ese sentido, para garantizar que la información contenida en los bloques sea inmodificable se hace uso de dos herramientas criptográficas: *firmas digitales* y *algoritmos de hash*. Las *firmas digitales*, se dividen en dos grandes grupos: *claves públicas* y *claves privadas*. Desde una óptica informática, la *clave pública* se utiliza por los usuarios de la red para cifrar los mensajes que se enviarán a un nodo particular, mientras que la *clave privada* permite que un nodo lea o descifre dichos mensajes. Por lo tanto, se utilizan dos claves diferentes: una para cifrar y otra para descifrar. En la práctica, la *clave privada* se utiliza para aprobar las transacciones en la *blockchain*, mientras que la *clave pública* sirve para direccionar cada nodo en la red [16]. Los *algoritmos de hash*, en tanto, sirven para obtener la "huella digital" de cada bloque a fin de que se pueda verificar que el contenido de este no ha sido alterado (ver *Figura 2.2*). Como se ha dicho, el *hash* faculta la concatenación, por lo que, ante la más mínima variación del contenido de un bloque, el código alfanumérico de la "huella digital" cambiaría drásticamente, imposibilitando la conexión entre los bloques e invalidando las transacciones de la *blockchain* (ver ítem 8.2.1 en Anexos). En términos prácticos, esto último implica que una vez que un *minero* consigue un *hash*, el bloque

minado no puede modificarse a menos que se reconstruya toda la cadena desde el *génesis* hasta dicho punto de ensamble y antes de que la red pueda procesar el siguiente bloque legítimo [5].

- **Información distribuida y no falsificable:**
  El hecho de que cada computador de la red distribuida tenga una réplica exacta y actualizada de la *blockchain* permite que todos los usuarios de la red estén al tanto de las nuevas transacciones del sistema (ver *Figura 2.1*). Además, como explica J. Todó [7], en caso de que un usuario no esté operativo, la información puede ser enviada a los demás participantes por vías alternativas conectadas a dicho punto. El hecho de que la base de datos no esté centralizada en un único servidor hace menos vulnerable el sistema y permite disminuir radicalmente las probabilidades de fraude, puesto que para efectuar transferencias falsas sería necesario modificar la *blockchain* en la mayoría de los nodos de la red; específicamente por encima del 51% del total de participantes [13] y antes de que se consolide la adición del bloque entrante (lapso de 10 minutos para *Bitcoin*, por ejemplo). Así, gracias a la concatenación secuencial de los bloques basada en el *consenso* de la red la *blockchain* es inmutable (ver *Figura 2.2*), y con ello, tener éxito en cursar una transacción falsa en cualquier estado del sistema y sobre todos los nodos de la red resulta muy difícil.

- **Conservación del histórico de datos:**
  Como se muestra en la *Figura 2.2.*, *blockchain* permite añadir bloques estampando una *marca de tiempo* del estado del sistema (*timestamp*) en cada uno de ellos, sin que sea posible retirarla. Esto último, más las características vistas con antelación, convergen en que esta base de datos distribuida se conciba como un registro cronológico de las transferencias efectuadas en la red, ofreciendo a los usuarios la trazabilidad instantánea del *activo digital* desde el bloque *génesis* hasta el último bloque añadido a la *cadena*, con un nivel de transparencia e integridad superior.

De las principales características enunciadas en este ítem se desprende que *blockchain*, entendida como un registro de datos distribuido, exhibe el historial inmutable de eventos que ha tenido un *activo digital* desde su origen, sin que sea necesaria la intervención de una tercera entidad que audite los procesos, desplazando la validación centralizada tanto humana como artificial y sustituyéndola por una comunidad de observadores encargados de certificar y replicar las transferencias del sistema en tiempo real.

2.1.2.3 Ciclo de encadenamiento de bloques

En virtud de las bases de funcionamiento y de los rasgos diferenciadores de una red con *blockchain* (vistos en los dos ítems que anteceden), los pasos a seguir para añadir un bloque a la *cadena* son:

1. Los usuarios interactúan con la *blockchain* mediante *firmas digitales* expresadas en *claves públicas* y *privadas*. Utilizan la *clave pública* para autoasignarse una *dirección* en la red y la *clave privada* para firmar las transacciones. Si se ha firmado la transferencia, la información está lista para ser propagada y cualquiera de los participantes de la red distribuida puede comprobar la identidad del nodo difusor, quien, a través de su *clave privada*, se hace cargo de la veracidad de la transacción entrante [6].

2. Los nodos vecinos o próximos al punto de emisión se aseguran de que toda "transferencia entrante" sea válida antes de seguir difundiéndola, esto es, comprueban que la firma del emisor sea correcta *(clave privada*) y que este tenga suficientes "fondos" de *activo digital* para hacer el movimiento. Las transacciones verificadas se llevan a un lugar de almacenamiento (llamado *transaction pool*) en donde esperan ser añadidas a la *cadena*, mientras que las transferencias inválidas se descartan [14].

3. Las transferencias que han sido recopiladas y aprobadas mediante el proceso anterior se ordenan y empaquetan en un bloque candidato con cierta *marca de tiempo*. Como se ha mencionado, este proceso se llama *minería* y es cuando alguno de los nodos de la red (nodo validador o *minero*) consigue "sellar" este nuevo bloque mediante la asignación del *hash* correspondiente. La elección de los *mineros*, así como los contenidos de cada bloque, dependen del *protocolo de consenso* que emplea la red [6].

   Obs: Los conceptos *minería* y *protocolo de consenso* se profundizan en la sección Anexos, en el ítem 8.2.1.

4. Todos los nodos de la red comprueban que el "bloque candidato" presentado por un *minero* es coherente a la *blockchain* inspeccionando si las transacciones son válidas y si el *hash* propuesto se vincula con el del bloque anterior. En caso afirmativo, agregan el bloque a la cadena y aplican las transacciones que contiene este; dejando constancia de la incorporación del mismo a todos los participantes de la red (*consenso*). Si ese no es el caso, el bloque propuesto se descarta. Bajo cualquier escenario, esto marca el final de un ciclo y la red está lista para evaluar a los nuevos bloques postulantes.

2.1.3 Taxonomías

Hay varias formas de categorizar una red de *blockchain*. De acuerdo con lo estudiado en las fuentes [15], [16] y [17], las clasificaciones se centran en la gestión de los datos, la disponibilidad de estos y el rango de acción que tiene un usuario; por lo que una primera aproximación permitiría discriminar entre *blockchains públicas*/*privadas* y *blockchains con permisos*/*sin permisos*. Una propuesta de clasificación de distintos tipos de *blockchains* se ilustra en la *Figura 2.3*, donde se muestra un resumen gráfico hecho por T. Fernández y P. Fraga [16] basado en las características más distintivas de algunas plataformas distribuidas y descentralizadas.

Según un trabajo publicado en la *IEEE Transactions on Knowledge and Data Engineering* [15], en una *blockchain pública* cualquier nodo puede unirse y abandonar el sistema sin la aprobación de terceros, por lo que la red está completamente descentralizada, asemejándose más fielmente a un sistema *peer to peer*. A su vez, estas redes suelen ser transparentes y los usuarios son anónimos [17], pudiendo actuar como un nodo simple o como un *minero* (validador). En las *blockchains públicas* a los nodos validadores se les suele dar incentivos económicos para mantener la inmutabilidad de la red, como por ejemplo en *Bitcoin*, *Ethereum* o *Litecoin* [16].

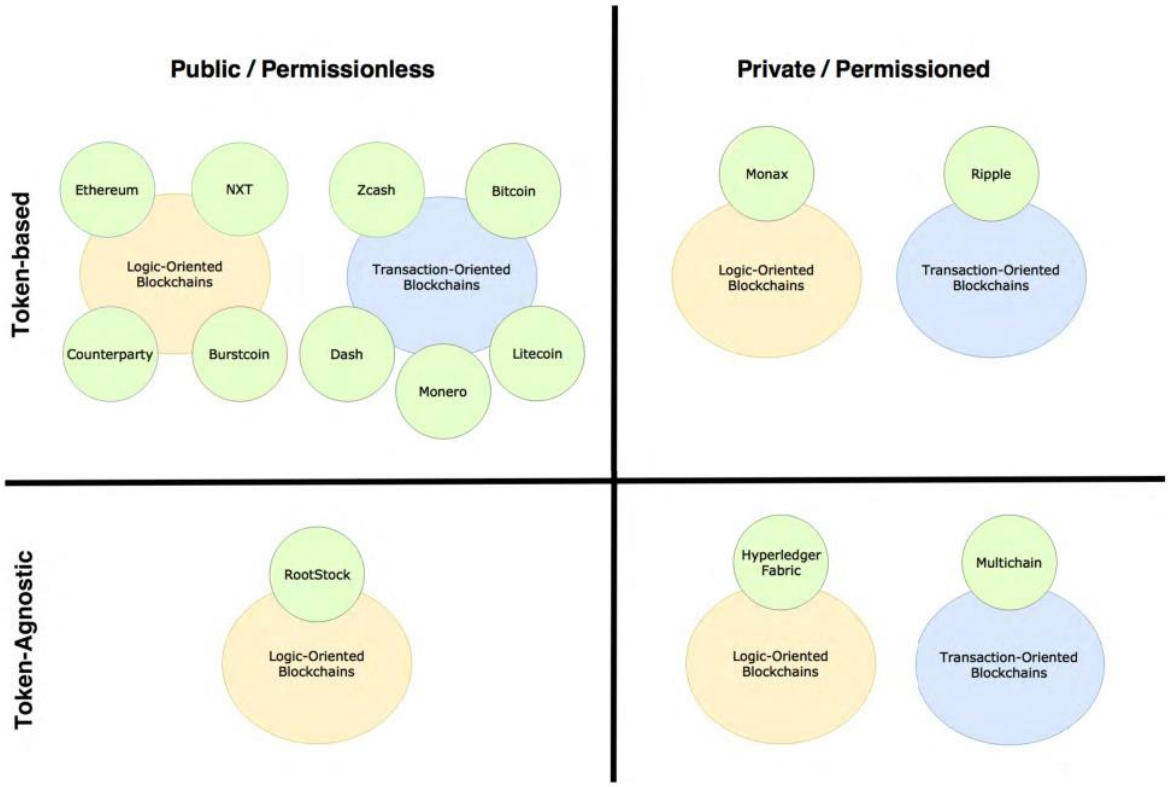


Figura 2.3: Taxonomía *blockchain* según el artículo "A review on the use of blockchain for the Internet of Things" [16].

Por su parte, en una *blockchain privada* existe un mecanismo de control de acceso o *permiso* para determinar quién puede unirse al sistema. Como se explica en el blog "Abierto al Público" del *Banco Interamericano de Desarrollo* (*BID*) [17], en estas redes el control se reduce a una única entidad que se encarga de mantener la *cadena*, dar *permisos* a los usuarios predeterminados, proponer transacciones y sellar los bloques. Ejemplos de *blockchains privadas* en las que se necesitan *permisos* para operar son *Hyperledger-Fabric*, *Parity* o *Sawtooth* [15]. Sin desmedro de lo anterior, es bueno comentar que, tal como enfatizan T. Fernández y P. Fraga en su paper [16], una organización puede implementar una *blockchain privada* basada en una *blockchain pública* como *Ethereum*, forzando a que dicha red opere *sin permisos*. Por tanto, utilizar como sinónimos los binomios *público-sin permisos* y *privado-con permisos* no es del todo correcto.

Las *blockchains híbridas*, *federadas* o *dedicadas*, en tanto, representan la alternativa intermedia entre *blockchains públicas* y *privadas*. Algunas de las plataformas que se utilizan para crear este tipo de redes son *Hyperledger-Fabric*, *Stellar*, *Corda* y *Multichain*. Comúnmente utilizan *protocolos de consenso híbridos* y, en general, no utilizan incentivos para el proceso de *minado* [6], siendo las instituciones, entidades o compañías quienes proporcionan los recursos tecnológicos necesarios para cumplir con la seguridad, integridad e inmutabilidad de la red. Consistentemente, según aclaran los expertos en el blog del *BID*, "el acceso mayoritario es mediante una interfaz web que los administradores ponen a disposición del usuario medio, en lugar de compartirles una copia de la cadena como en las redes *públicas*" [17].

Para finalizar, y en referencia al esquema de la *Figura 2.3*, se observa que no existe una única forma de clasificar a la tecnología *blockchain*, dando la posibilidad de generar más categorías. Entre ellas, están las plataformas orientadas al intercambio de criptomonedas (como *Bitcoin*) y las *blockchains* facilitadoras de instrucciones lógicas (como los *smart contracts* promovidos por *Ethereum*, por ejemplo); así como también las plataformas que transan *tokens* (como en *Ripple*) y las que no (como *Multichain*) [17]. Otra característica que permite encasillar las variantes de esta tecnología es el *protocolo de consenso* empleado para validar las transacciones. De acuerdo con lo que se expone en el ítem 8.2.1 de la sección Anexos de este trabajo, la *minería* de bloques se puede ejecutar con *protocolos basados en cómputo* (como la *proof of work* usada en *Bitcoin* o la *proof of stake* ocupada en *Ethereum*), *protocolos basados en comunicación* (como el *protocolo practical byzantine fault tolerance* en *Hyperledger-Fabric*) y *protocolos híbridos* (como la *proof of elapsed time* utilizada para optimizar la *minería* en *blockchains públicas*, o bien, la *proof of authority* empleada para eliminar los incentivos económicos del *minado* en *blockchains públicas* y *privadas*) [15][17].

2.2 Smart Contract: una poderosa aplicación de blockchain

2.2.1 Definición

En términos generales, un *smart contract* o *contrato inteligente* es un programa informático alojado en una red con *blockchain* que facilita el cumplimiento de un acuerdo entre dos partes a través de la ejecución de instrucciones lógicas previamente definidas en el sistema distribuido [15][16].

En el documento *Blockchain and smart contracts for the internet of things* [6] se atribuye la autoría del concepto a Nick Szabo, quien durante 1994 tuvo la idea de adaptar cláusulas contractuales en un software para que, mediante su ejecución, este pudiera hacerlas cumplir. El propósito era disminuir las faltas o fraudes en los acuerdos y minimizar la necesidad de intermediarios de confianza entre las partes. Así, según se cita en [6], N. Szabo definió un *smart contract* como "un protocolo de transacción computarizado que ejecuta los términos de un contrato".

2.2.2 Funcionamiento

2.2.2.1 Descripción básica

Para llevar a cabo las instrucciones lógicas que le competen, un *smart contract* puede verse como un procedimiento almacenado en un nodo de la red que se invoca según la necesidad de los usuarios. De este modo, las entradas, salidas y estados resultantes de la interacción entre un nodo y el *contrato* equivalen a transacciones que deben ser consensuadas por todos los participantes del sistema [15].

Según explican los autores de [6], un *smart contract* tiene una *dirección* única en la red (*clave pública*), lo que facilita el envío de *activos digitales* para ejecutar las instrucciones. Esto se hace de forma independiente y automática con un grupo de nodos convenidos dentro del sistema.

Conforme plantea [15], todas las *blockchains* tienen incorporados algún tipo de *smart contract* para efectuar sus lógicas de transacción. Por ejemplo, para operar con criptomonedas, debe existir un *contrato* para la verificación de *firmas digitales* asociadas al emisor y receptor del *activo* (*claves pública* y *privada* de los respectivos agentes). Este comprobará que las cantidades transadas de entrada y salida son coherentes (piénsese en saldo de *activos digitales*, suma a enviar, monto a recibir, etc.) y, en último lugar, aplicará el cambio de estado en las cuentas del sistema estampándoles la *marca de tiempo* que las diferenciará del resto de las operaciones, generando nuevos saldos de *activo* que pueden ser potencialmente transferidos a otros nodos de la red.

2.2.2.2 Principales características

El hecho de que un *smart contract* opere con tecnología *blockchain* le reporta características diferenciadoras sobre los contratos tradicionales, siendo la ejecución autónoma y cíclica de las reglas entre los agentes de la transacción las más relevantes.

Como nodo de la red, las transferencias desde y hacia un *smart contract* heredan, también, las características de la *blockchain*, a saber: *consenso*, *concatenación*, *inmutabilidad*, *información distribuida y no falsificable* y *conservación del histórico de datos* (ver ítem 2.1.2.2 de este capítulo). Esto facilita que las reglas que rigen la transferencia de activos se efectúen prescindiendo de una autoridad central, pues, en su lugar, la validación del *contrato* se hace con múltiples observadores.

2.2.2.3 Interacción con smart contract

Para comprender mejor cómo funciona un *contrato inteligente* se utilizará una interpretación del ejemplo presentado por K. Christidis y M. Devetsikiotis en su trabajo de 2016 [6]. En este ejemplo se tiene una red con *blockchain* en la que existen dos usuarios: Alice y Bob. En dicho escenario, los participantes desean intercambiar *activos digitales* de tipo x e y mediante transacciones descentralizadas entre sus cuentas (nodos).

El paper [6] dice que Bob programa un *smart contract* que despliega en la red y en el cual define funciones que sólo él puede invocar (que viene a ser una condición particular para este ejemplo) y que se describen a continuación:

- Función "depósito": permite que Bob pueda depositar unidades de x en el contrato.
- Función "comercio": por cada 5 unidades de y recibidas en el contrato, se envían 2 unidades del *activo digital* x a la cuenta del remitente.
- Función "retiro": permite que Bob retire el total de *activos* que posee el *contrato*.

Cuando Bob envía una transacción a la *dirección* de dicho *smart contract* para sumar $5x$ a los “fondos de activo” que viven en el *contrato*, este invoca a la función "depósito" y adiciona el monto pactado. Naturalmente, esta operación queda registrada en la *blockchain* adoptada en la red.

Alice, por su parte, quien posee 12 unidades de y en su cuenta, envía una transacción de $10y$ a la función de "comercio" y, como consecuencia, obtiene $4x$ de vuelta. Dicha transacción también se registra en la *blockchain*.

(a)

Propietario/a	Tipo de activo	Cantidad
Bob	x	10
	y	0

Propietario/a	Tipo de activo	Cantidad
smart contract	x	0
	y	0

Propietario/a	Tipo de activo	Cantidad
Alice	x	0
	y	12

$\Rightarrow$

(b)

Propietario/a	Tipo de activo	Cantidad
Bob	x	5
	y	0

Propietario/a	Tipo de activo	Cantidad
smart contract	x	1
	y	10

Propietario/a	Tipo de activo	Cantidad
Alice	x	4
	y	2

Tabla 2.1: Ejemplo de transferencia de *activos digitales* vía *smart contract* basado en *blockchain*. **(a)** Estado inicial de todas las cuentas o nodos implicados. **(b)** Estado de todos los nodos implicados antes de transferir a Bob los *activos* que posee el *contrato*. Interpretado de [6].

Lo que sigue en el ejemplo de los autores [6] es que Bob envía una transacción *firmada* a la función de "retiro" para hacer uso de los *activos* acumulados. En este caso, el *smart contract* verifica la *clave privada* de Bob para garantizar que el retiro será efectuado por el propietario de la cuenta y, una vez comprobada, transfiere todos los depósitos a la *dirección* de Bob (a saber: $1x$ y $10y$).

Como puede notarse en la *Tabla 2.1*, las cuentas de los usuarios han mutado con la interacción, surgiendo varios aspectos interesantes. Entre ellos se aprecia que el *smart contract* tiene la calidad de nodo en la red, ya que, al igual que las cuentas de Bob y Alice, este también tiene los campos “Tipo de activo”, “Propietario/a” y “Cantidad”, confiriéndole la calidad de “usuario”; vale decir, facultándolo para poseer y modificar saldos en la *blockchain*. Además, dado que el *contrato* reside en la red distribuida y debido a que todas las interacciones conllevan la *firma digital* de los participantes, cualquier nodo de la red puede consultar el histórico de las operaciones [6].

También, dado que el *smart contract* de Bob tiene su propia *dirección* en la red, la ejecución de las “reglas de intercambio” entre la cuenta de Bob y la de cualquier otro remitente se automatiza, dejando un registro cronológico del cumplimiento de las cláusulas o condiciones programadas.

Adicionalmente, una última observación relevante es que un *contrato inteligente* es “determinista”. Esto quiere decir que, si el archivo recibe constantemente la misma entrada, este siempre producirá

la misma salida. En caso contrario, si se escribiera un *smart contract* con instrucciones "no deterministas", la activación del código mermaría en diferentes resultados de tipo aleatorio en cada nodo del sistema, lo que dificultaría el *consenso* de la red [6].

Con el ejemplo revisado, se pretende que el lector distinga que una *blockchain* que acepta la incorporación de *smart contracts* lleva a la contabilidad distribuida de *activos digitales* por encima de las aplicaciones de las criptomonedas, facilitando todo tipo de procesos que se valen de condiciones lógicas para controlar la desconfianza entre usuarios, pudiendo estos [6]:

- Identificar los resultados de la aplicación de un *smart contract* antes de comprometer obligaciones con la contraparte (gracias a la inspección previa del código).
- Tener certeza de la imparcialidad del código ejecutado (ya que el *smart contract* está implementado sobre una red en la que ninguno de los agentes de la transacción influye).
- Verificar la totalidad del proceso, permitiendo auditar el historial de las transferencias (pues todas las interacciones están con *firma digital* y *marca de tiempo*).

Finalmente, es bueno mencionar que la disputa respecto de los resultados de un *smart contract* no debiese existir, dado que el código del archivo puede ser analizado con antelación en entornos de desarrollo de programación (denominados "IDE") para comprobar que las salidas son plenamente predecibles o "deterministas", por lo que ambos agentes de la transacción pueden verificar y simular los movimientos de la contraparte [6].

2.3 Aproximación de blockchain al sector energético

Conforme se expone en el documento *Profiles in Innovation: Blockchain, putting theory into practice* del grupo de Banca e Inversiones *Goldman Sachs* [20], la actual operación del sistema eléctrico está basada en un grupo reducido de entidades encargadas de gestionar amplios volúmenes de información en un mercado relativamente cerrado. Al respecto, entendiendo al parque generador como un agente sumergido en un escenario de competencia perfecta y considerando el constante aumento de la demanda, es natural esperar un crecimiento en la inversión de tecnología ubicada lejos de los centros de consumo e incrementos de gasto de capital en infraestructura de transmisión y distribución, con lo cual la tarifa eléctrica de los clientes finales también experimenta alzas.

Más aún, a lo largo de la *cadena de suministro* de la electricidad existen diversos agentes, a saber: empresas de generación, empresas de transmisión, empresas de distribución y de servicios; que

cumplen el rol de intermediarios entre la producción y el consumo de energía eléctrica en su formato tradicional, acrecentando los costos de suministro tanto por razones técnicas (pérdidas asociadas a transporte) como comerciales (uso de las instalaciones, tipo de cliente, tipo de tarifa, etc.).

Al mismo tiempo, la conciencia medioambiental y las obligaciones de los países frente al cambio climático, la consolidación de tecnologías de generación con energías renovables, el almacenamiento energético, la electromovilidad, la irrupción del *internet de las cosas* (*IoT* por sus siglas en inglés) y las exigencias ciudadanas por mayor transparencia en los procesos energéticos, está empujando a las empresas e instituciones a reemplazar el modelo actual por un modelo más descentralizado y propositivo.

En ese sentido, y en coherencia a lo que plantea J. Todó en su tesis [7], se prevé que el parque generador experimente cambios debido a que se potenciará la generación distribuida (acercando las inyecciones de energía a los centros de consumo) y, al mismo tiempo, la demanda tomará un rol mucho más activo en el mercado energético, por cuanto los clientes finales podrán aportar a la operación de las futuras redes con la gestión de su propio consumo.

En este escenario, como advierten J. Schneider, A. Blostein *et al.* [20], la introducción de tecnologías como *blockchain* representa una oportunidad para concretar el cambio hacia una red eléctrica más distribuida, donde será de vital importancia contar con *smart contracts* y "medidores inteligentes" que permitan la gestión automática de las transacciones energéticas. De este modo, se podrá reducir la vulnerabilidad de los datos de energía, además de aminorar los costos de burocracia que suponen eventuales auditorías o trámites con instituciones validadoras, fomentando oportunidades de negocio entre productores, consumidores y la nueva figura incipiente del mercado: los "prosumidores", quienes consumen y producen electricidad bajo el esquema de generación distribuida.

También, debido a que la *blockchain* enseña las relaciones de propiedad de un activo transferido y la *marca de tiempo* de cada uno de los movimientos, cualquier participante de la red podrá hacer una retrospectiva (leer la cadena "hacia atrás") y averiguar el estado de los datos transados en la red distribuida [6]. De manera que *blockchain* no sólo será atractivo para ejecutar un intercambio comercial directo o *peer to peer* en la red eléctrica, sino que también resultará útil en el seguimiento de la *cadena de suministro* de la electricidad y como prueba de procedencia de una inyección o retiro en un punto específico del sistema.

En coherencia a lo anterior, y según expone F. Orecchini, A. Santiangeli, *et al.* en [25], la utilización virtuosa de recursos energéticos disponibles en una *red inteligente* puede concretarse a través de dispositivos que consideren la observación, reconocimiento y cuantificación de las inyecciones en el sistema eléctrico. Al respecto, los autores sostienen que la "trazabilidad de la energía" en un contexto de creciente sustentabilidad es una herramienta de desarrollo importante, porque sólo de esta manera es posible conocer el origen exacto de la electricidad utilizada para un uso final específico, como la carga energética de vehículos eléctricos, por ejemplo [25]. De este modo, el uso de *blockchain* puede ser de gran ayuda para el rastreo de historiales reales de intercambio de energía entre generación y demanda, además de la determinación de la huella de carbono tanto de la matriz energética convencional como de las tecnologías entrantes.

Papers, revistas, informes técnicos y una serie de publicaciones de parte de instituciones como *IEEE*, *BID*, *IBM*, *BBVA*, *Morgan & Stanley Co.* y la propia *Comisión Nacional de Energía* en Chile, son sólo una muestra del interés que hay en torno a esta temática. En el sector de la educación superior iberoamericana, por ejemplo, la *Universidad de Sao Paulo* y la *Pontificia Universidad Católica de Chile* han comenzado a dictar diplomados, mientras que la *Universidad Politécnica de Cataluña* ya imparte su primer máster en *smart energy*, donde la tecnología *blockchain* y las energías renovables ocupan un eje importante.

Respecto de la implementación de *blockchain* en el sector Energía, algunas de las iniciativas alrededor del mundo que se encuentran actualmente desarrollando proyectos con diferentes grados de avance, son:

- ***Slock.it***

  Durante 2016, esta empresa alemana desarrolló un sistema *peer to peer* en *Ethereum* para que distintos usuarios puedan compartir sus estaciones privadas de carga de vehículos eléctricos. El proyecto *Share & Charge*, por medio de una aplicación web, facilita que los conductores que necesiten cargar sus autos conozcan cuáles son los puntos de carga o "polos" de energía más cercanos [30]. Luego, una vez que el vehículo está enchufado a la red domiciliaria del "polo" escogido, el usuario paga sólo por la energía consumida. El dueño del punto de carga, a su vez, recibe una recompensa a través de *tokens* que pueden canjearse por diversos servicios que ofrece la plataforma *online* o por dinero en efectivo [30].

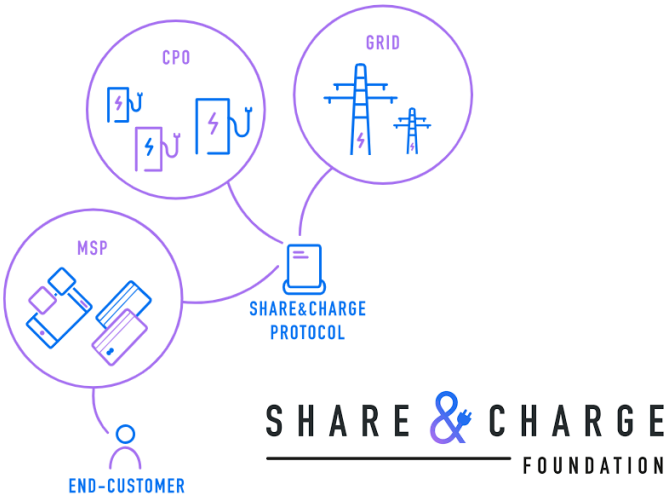


Figura 2.4: Infografía simple del proyecto de carga para vehículos eléctricos de la empresa *Slock.it* [30].

- ***WePower***

  Con el fin de potenciar las energías renovables en Lituania, esta empresa propuso un sistema de financiamiento para "productores verdes" a través de la emisión de *tokens*. La idea es que cada proyecto adquiere un capital inicial mediante la compra de *tokens* de parte de los consumidores.

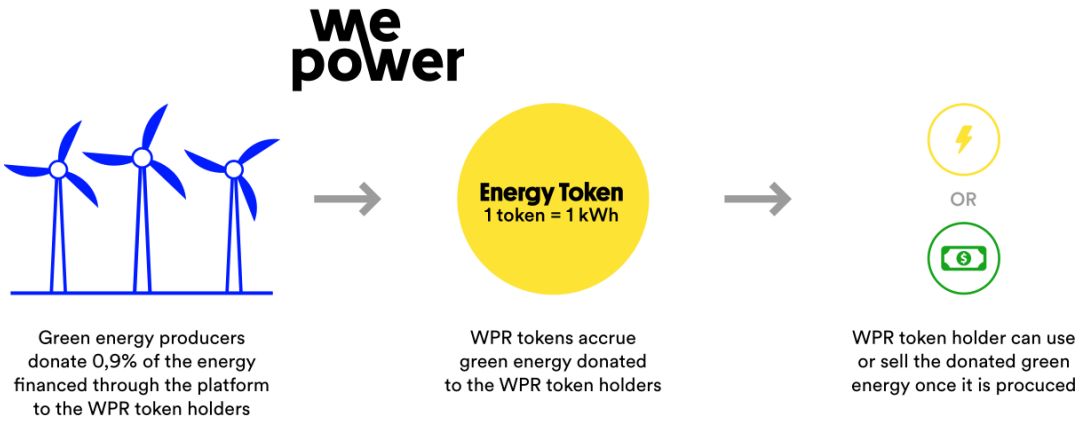


Figura 2.5: Infografía simple de iniciativa *WePower*. Presencia en Lituania, Estonia, Australia y España [31].

Al mismo tiempo, cada *token* representa un kilowatt-hora de energía que los generadores se comprometen a inyectar una vez concretado el proyecto, por lo que, desde la etapa de inversión y durante la ejecución de las plantas, existe una interacción comercial directa entre los productores, consumidores e inversionistas. La "*tokenización* energética" de

WePower, por tanto, permite a los generadores renovables obtener capital gracias a la venta de energía por adelantado a tasas más bajas que las de la Banca tradicional, a la vez que los consumidores acceden a un precio de electricidad inferior al que ofrece la empresa distribuidora [31].

- ***Klenergy Tech***

  Esta empresa española, con sede también en Dinamarca, se enfoca en la descentralización del mercado eléctrico mediante la promoción de una plataforma *online* llamada *Pylon Network*, diseñada para el intercambio *peer to peer* de energía renovable proveniente de plantas de diversos tamaños (incluida la generación distribuida). La comercialización se sustenta en recompensas para los generadores y en herramientas de gestión de demanda para los consumidores. La iniciativa ejecuta los *smart contracts* en *Ethereum* para garantizar transferencias seguras, rastreables y en ausencia de intermediarios. Por su parte, el *minado* de bloques se premia con una criptomoneda llamada *Pylon Coin*, mientras que el financiamiento colectivo para eventuales proyectos se realiza con *Pylon tokens* [32].

Con lo anterior, resulta interesante notar que, si bien ya existen iniciativas en marcha como "The Brooklyn Microgrid – P2P energy trading" [34] desarrollado por la empresa ***LO3 Energy*** en conjunto con *Siemens* en Nueva York (donde el comercio de electricidad entre vecinos gestionado a través de *blockchain* permitirá pasar de un modelo de energía tradicional a uno mucho más comunitario y sostenible), todavía se deben adoptar cambios en la regulación y en la infraestructura de la red eléctrica para potenciar estas iniciativas a escala local o global.

Figura 2.6: Microgrid ubicada en "President Street", NY. El proyecto pretende crear un *pool* de energía entre prosumidores y consumidores, con el fin de que cada uno elija a su proveedor dentro de una red local [34].

3. Sistema de trazabilidad de energía basado en blockchain

Teniendo en cuenta los conceptos generales relativos a trazabilidad, *blockchain* y *smart contracts*, abordados en las secciones anteriores de este trabajo, este capítulo pretende mostrar los rasgos más esenciales del sistema de trazabilidad de energía eléctrica formulado por la empresa *Phineal* [23].

3.1 Blockchain GTIME: el blockchain de energía

El sistema de trazabilidad de energía eléctrica desarrollado por la empresa *Phineal* se sustenta en una *cadena de bloques federada* llamada *Blockchain GTIME* [26], que es un registro inmutable de transacciones de energía eléctrica, consensuado y mantenido por una red descentralizada de computadores independientes. Cada transferencia energética sirve como prueba de procedencia de los flujos inyectados o retirados en un punto específico del sistema eléctrico, permitiendo la trazabilidad cronológica de estos mediante la consulta "online" del histórico de movimientos.

Figura 3.1: Logo de la iniciativa *blockchain* de energía llevada a cabo por *Phineal* [23][26].

Para medir las variables involucradas de manera empírica, la iniciativa cuenta con dispositivos electrónicos denominados *phiNet* y *phiNergy*, que se instalan directamente en las plantas de generación y/o centros de consumo convenidos para el registro de la electricidad producida o demandada [23][26], y que corresponden a los nodos de la red con *blockchain*.

Dichos dispositivos están diseñados para agrupar y emitir, en un lapso de 15 minutos, un paquete de información que contiene la ubicación georreferenciada del medidor, el tiempo exacto en el que se ha sensado la energía, la *ID* del usuario, la *MAC Address* del dispositivo y la cantidad de energía eléctrica que ha sido medida [23][26]. A este vector de información se le denomina "*GTIME*" en respuesta a las iniciales de las cinco variables antes mencionadas y se simboliza mediante el pentágono que se ilustra en la *Figura 3.2*.

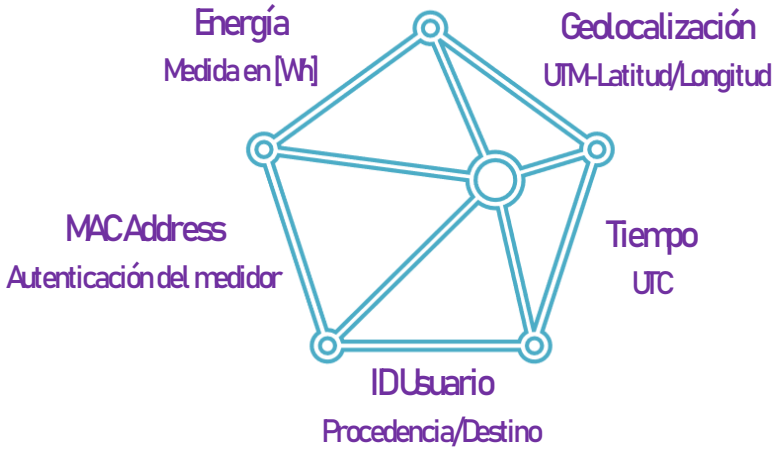


Figura 3.2: Ícono del vector "*GTIME*", que constituye el *activo digital* a transar en la *Blockchain GTIME* de *Phineal* [26].

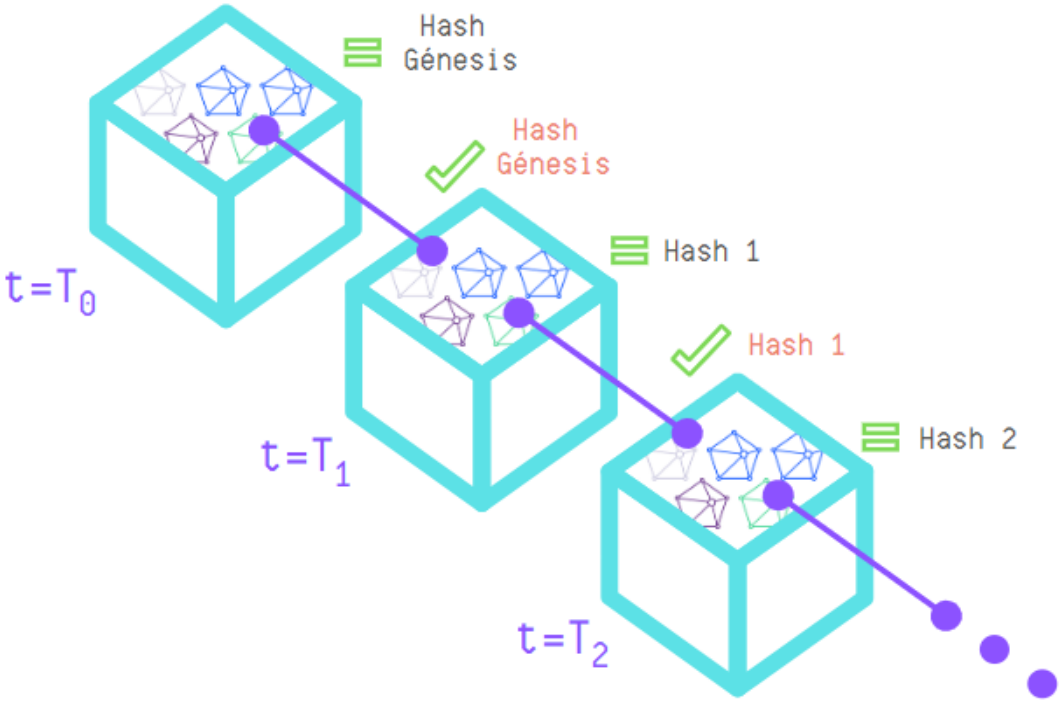


Figura 3.3: Esquema de concatenación de bloques para las transacciones de vectores *GTIME* [26]. Elaboración propia.

Dentro de los 5 minutos siguientes, los dispositivos establecen una comunicación *machine to machine* (M2M) vía internet con la plataforma *blockchain* de *Phineal* para depositar los datos reclutados [23]. Por tanto, en un intervalo de tiempo de 20 minutos, el conjunto total de equipos *phiNet* y *phiNergy* captura, registra, ordena y empaqueta la información energética del sistema eléctrico para luego enviarla a la *Blockchain GTIME*, en donde se "estampa" la *marca de tiempo*

sobre el grupo de vectores y, después de ello, se procede a crear un bloque [26]. Posteriormente, y en coherencia al ítem 2.1.2 del Capítulo 2, el bloque se somete a un proceso de *minería* para la asignación del *hash*, lo que permite ratificar las transacciones entrantes e incorporarlas a la *cadena de bloques* (ver *Figura 3.3*). Con ello, se alcanza el *consenso* global de los dispositivos validadores, haciendo posible la trazabilidad temporal de las inyecciones o retiros, según sea el caso.

Compuesta por una plataforma web que muestra las transferencias de electricidad en tiempo real de manera segura y verificable, la *blockchain federada* de *Phineal* da pie a los siguientes servicios:

- **Trazabilidad energética**

  Gracias a la medición *in situ* de las variables pertinentes al protocolo *GTIME*, sumado a la concatenación cronológica e inmutable de los bloques de información, es posible responder a todas las preguntas de la procedencia energética, tales como: ¿Dónde se genera/consume la energía? ¿En qué instante se registró la energía? ¿Quién inyecta/retira la energía? y ¿Cuánta energía está en juego? [23][26]. Con lo anterior, se debe entender que *Blockchain GTIME* puede desglosar el servicio de trazabilidad energética en dos categorías:

  <u>Trazabilidad de la producción energética</u>: en cuyo caso se rastrean las inyecciones de energía que sirven como prueba de procedencia de la energía utilizada. Este servicio es ideal para empresas generadoras que deseen incorporar certificados de origen de energías renovables o cuantificación de sus inyecciones al sistema. Los clientes libres o regulados que cuentan con instalaciones de generación distribuida para autoconsumo también pueden acceder a este tipo de servicio.

  <u>Trazabilidad de la cadena de suministro:</u> donde, mediante un *smart contract*, se realiza la cuantificación física de las inyecciones, retiros y pérdidas eléctricas involucradas en el proceso (las que pueden calcularse mediante la importación de datos relevantes alojados en plataformas como la del Coordinador Eléctrico Nacional, Comisión Nacional de Energía o de cualquier otro organismo pertinente). Este servicio de información física (piénsese en $[Wh]$ ó $[W]$) es ideal para certificados de abastecimiento de energías renovables en industrias y organizaciones que ya poseen "suministro verde" a larga distancia (ya sean clientes libres o regulados) y que desean construir una contabilidad auditable de las emisiones de carbono de sus productos y/o servicios. Los puntos de carga de vehículos eléctricos también podrán beneficiarse de este tipo de certificados para promover recargas verdaderamente "limpias".

- **Smart PPA**

  La programación de *smart contracts* que incluyan cláusulas relativas a contratos de suministro eléctrico en los que se utilicen los atributos de la *Blockchain GTIME*, además de la dinámica de precios asociada al intercambio energético, permitirán autoejecutar las condiciones pactadas y verificar, en todo momento, el cumplimiento de los acuerdos entre la generación y el consumo. En este servicio se pueden incorporar las inyecciones, retiros y pérdidas asociadas al transporte de la energía, así como la valorización de las mismas, subyugando las instrucciones de código de programación al marco legal y regulación vigente. Este mecanismo se adapta bien a clientes libres y regulados que deseen establecer un contrato de compra/venta de electricidad o *PPA* (siglas del inglés *Power Purchase Agreement*) con generadoras renovables a un precio más conveniente que el que ofrece la compañía de transporte [23][26]. También, sirve para gestionar la valorización auditable de los movimientos energéticos de prosumidores y consumidores en un proyecto colaborativo de generación distribuida (como "The Brooklyn Microgrid" [34], por ejemplo); aunque, para estos fines, aún faltan avances en la regulación.

- **Plataforma de marketing**

  Gracias al etiquetado de códigos QR dinámicos en productos y servicios, los clientes finales pueden acceder a una plataforma web que valida los datos energéticos utilizados para su creación, promoviéndolos como bienes con baja huella de carbono y gestados a partir de fuentes renovables [23][26]. Al mismo tiempo, las empresas e instituciones ven enriquecida la *cadena de valor* de sus procesos, por la externalidad positiva que ello implica.

Finalmente, en sintonía a lo publicado por la empresa en su web oficial, el desarrollo de *Blockchain GTIME* pretende encabezar la transición hacia un mercado centrado en los prosumidores, quienes podrán generar y/o almacenar su propia energía de manera distribuida, para luego venderla al resto del sistema de forma totalmente descentralizada [23]. En ese sentido, *Sello Sol* representa una de las aplicaciones pioneras de *Phineal* en Latinoamérica, y está enfocada en la certificación de productos y servicios gestados a partir de la energía solar; cuya trazabilidad energética se lleva a cabo con *Blockchain GTIME*. **Obs:** se ha presentado una pequeña reseña de *Sello Sol* en el ítem 8.3.3 de la sección Anexos.

3.2 Prototipo tecnológico del sistema de trazabilidad

Este ítem tiene por finalidad describir brevemente las cualidades más relevantes del *blockchain* de energía de *Phineal*, sobre todo en lo que tiene que ver con el software y hardware utilizado en las aplicaciones de trazabilidad eléctrica más recientes.

3.2.1 Taxonomía blockchain

Tal como se estudió en el Capítulo 2, hay muchas formas de categorizar a una *cadena de bloques*. En el ítem 2.1.3 se comentó que las categorías más evidentes se enfocaban en cómo se gestionaban los datos (de forma centralizada o descentralizada), en el acceso a la información transada (visualización pública o privada de los movimientos) y en el rango de acción que tiene un usuario de la red (si puede o no formar parte de la *blockchain* y en qué calidad; a saber: nodo, *minero* o mixto). En ese contexto, *Blockchain GTIME* puede entenderse como una *blockchain dedicada*, *híbrida* o *federada*. Una primera cualidad para catalogarla como tal es que *Phineal* es quien se encarga de seleccionar a cada uno de los nodos participantes mediante la entrega de un *permiso* que les confiere acceso a la red. En términos concretos, los nodos que componen la red distribuida pueden ser de dos tipos: medidores (*phiNergy* o con *vinculación phiNet*) y validadores (*Miners*). Dentro de este último grupo hay cabida para que existan computadores independientes externos a la propiedad de la empresa, pues son los nodos validadores quienes permiten descentralizar las transferencias del sistema eléctrico.

Una segunda razón para situar a *Blockchain GTIME* en el espectro intermedio entre *blockchains públicas* y *privadas* es que no existen incentivos económicos para el sellado de bloques. Si bien el *protocolo de consenso* utilizado es la *prueba de trabajo* o *proof of work* (*PoW*), la *dificultad de minado* es baja, lo que disminuye los costos computacionales implicados en la confirmación y mantención de la *cadena de bloques* (para refrescar conceptos diríjase al ítem 8.2.1 en Anexos). Esta idea implica que el conjunto de nodos validadores externos al grupo de *Miners* de *Phineal* no debe tener ánimos de lucro, como, por ejemplo: entes gubernamentales autónomos, fundaciones o cualquier otra institución que desee monitorear las cantidades de energía. Un último punto para categorizar a la *Blockchain GTIME* como *blockchain híbrida* es que la información de las transacciones se sube a una plataforma web, desde donde es posible consultar públicamente toda la información contenida en los vectores *GTIME* transferidos. Sin embargo, el acceso a una copia actualizada de la *cadena de bloques* sólo está disponible para aquellos nodos que pertenecen a la red.

Algo interesante a comentar es que, debido a la naturaleza de los datos con los que se trabaja en un SEP (entiéndanse magnitudes físicas como corrientes, tensiones, flujos de potencia, etc.) y dado que las leyes naturales que justifican la cuantía de dichas variables se basan en la medición y la observación (como la *ley de conservación de la energía*, por ejemplo), se infiere que en una *blockchain* de este tipo no hay espacio para la especulación de las cantidades transadas. Esto último sugiere que, a nivel físico al menos, la categoría *híbrida* de *Blockchain GTIME* no debiera decantar en conflictos de transparencia de información; justificando la selección centralizada de nodos únicamente por la calidad técnica de las mediciones, la eficiencia del levantamiento de datos y el cálculo de *hashes*, respectivamente.

3.2.2 Componentes básicos y funcionamiento del sistema GTIME

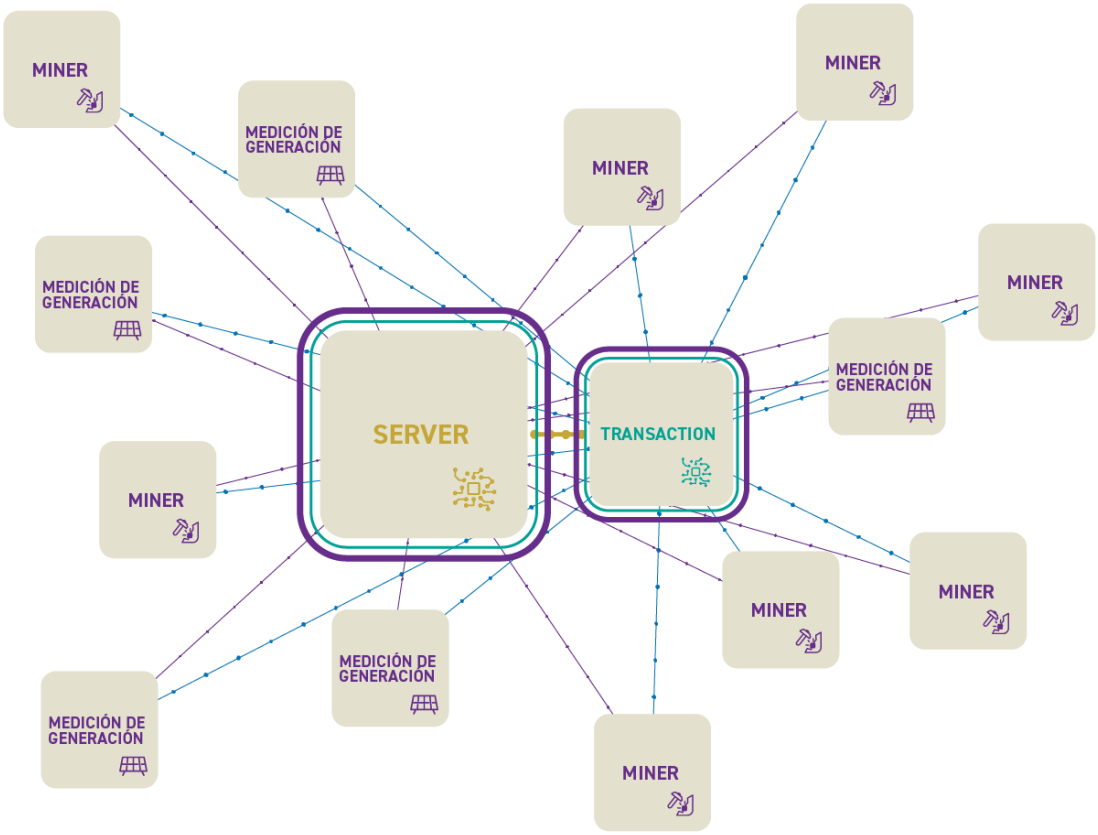


Figura 3.4: Esquema de composición de unidades de información en el *blockchain* de energía de *Phineal* [25][26].

A nivel conceptual, la distribución esquemática de los nodos en la primera versión de la red *Blockchain GTIME* es como se muestra en la *Figura 3.4*. Como podrá notarse, en esta primera versión de *Blockchain GTIME* se desprenden cuatro elementos fundamentales con roles claramente definidos:

- Medidores de generación/demanda
- Miners
- Server
- Transaction

Antes de explicar las funciones básicas de cada elemento, vale la pena reconsiderar la *Figura 3.4* para notar que en esta primera versión la red descentralizada se vale de un servidor único y que los nodos no están distribuidos, esto es, no están comunicados directamente o de forma *peer to peer*.

Lo que se hace, en su lugar, es que cada nodo envía un "update de status" al *Server* para mostrar la disponibilidad del medidor o validador (*online* u *offline*), y con ello, hacer un "request" o solicitud de estado de los nuevos bloques con el fin de que los nodos adquieran una copia actualizada de la *blockchain* (lo que permite descentralizar las transacciones energéticas).

Una vez confirmada la disponibilidad de cada medidor, se levantan los vectores *GTIME* almacenados y se llevan a la unidad *Transaction*, que es un mecanismo del *Server* dispuesto para que los nodos consulten las transacciones y el *minado* pendientes, con el fin de conformar un nuevo bloque. Esto último puede interpretarse como que las *phiNergys* y los medidores con *vinculación phiNet* "preguntan" al *Transaction* si el último bloque disponible está aún abierto para guardar allí los datos, mientras que los *Miners* "preguntan" al *Transaction* si el último bloque disponible está listo para ser sellado.

Si un bloque está listo para ser confirmado, los nodos medidores dejan de subir información y se dedican únicamente a capturar datos, mientras que los nodos validadores "compiten" para someter el último bloque hallado en el *Transaction* a un proceso de *minería*. Cuando el nuevo *hash* ha sido encontrado por uno de los *Miners*, este devuelve el bloque sellado al *Server*, para que lo incorpore a la *cadena* y pueda guardar y actualizar la versión de la *blockchain*. Finalmente, a través del siguiente "request", el total de nodos podrá obtener una nueva copia de la *cadena de bloques*, completando el ciclo de un proceso cuya duración actual ronda los 20 minutos, aproximadamente.

Por otra parte, conforme a lo que explica la empresa *Phineal* en [24] y [26], el código de *Blockchain GTIME* se ha diseñado para que en versiones posteriores se agreguen más servidores o, idealmente,

se adopte una arquitectura *peer to peer*. Este punto es relevante ya que, actualmente, todos los nodos de la red distribuida pueden "ver" que se actualiza su "libro mayor" a través del "request"; no obstante, el único elemento que almacena la información contenida en la *blockchain* (y con ello, el historial de transacciones) es el *Server*. Se debe entender, por tanto, que *Blockchain GTIME* pretende transitar hacia una topología distribuida en lo que a almacenamiento de datos refiere.

3.2.2.1 Medidores de generación/demanda

En esta primera versión del *Blockchain GTIME*, el registro empírico de los datos de energía se lleva a cabo por medidores de generación/demanda que se asignan dependiendo del tamaño del cliente, pudiendo dividirse en dos grandes tipos: *phiNergy* (pequeños clientes) y *red de estaciones phiNet* (grandes clientes).

3.2.2.1.1 phiNergy

Los medidores *phiNergy* son los equipos encargados de medir la producción o consumo de energía en el lugar mismo de la instalación para luego subir la información recopilada a través de internet (ver *Figura 3.5*). En la actualidad, estos medidores están diseñados para medir energía en baja tensión, por lo que su uso se orienta al segmento domiciliario, comercial y de pequeña industria.

Figura 3.5: Nodo medidor *phiNergy* [23].

Básicamente, la idea es que los datos que componen el vector *GTIME* se midan, se agrupen y se suban al *Server* para que, junto al resto de vectores informados por la red de *phiNergys*, se proponga un nuevo bloque al sistema (grupo sin *minar*, alojado en unidad *Transaction*). La ficha técnica de uno de estos medidores se comparte en la página web de la división *phiNet* [23][26], en donde la empresa *Phineal* publica los datos que se exhiben en la *Tabla A.3.1* (ver ítem 8.3.1 en sección Anexos, al final de esta Memoria).

En la práctica, un equipo *phiNergy* está compuesto por una tarjeta "Raspberry Pi Zero W" (ver *Figura 3.6*), una placa electrónica que permite la conexión de los sensores y una pinza amperimétrica no invasiva [23][26]. La tarjeta "Raspberry Pi Zero W" es una de las primeras versiones de minicomputador de tipo PCB (*printed circuit board*) diseñado por *Raspberry Pi Foundation* para trabajar como interfaz entre el software y el hardware que se desee [35]. En este caso, dicha PCB se utiliza para hacer "dialogar" las instrucciones del protocolo *GTIME* con el conjunto de sensores electrónicos acoplados en la *phiNergy*.

Figura 3.6: Tarjeta PCB "Raspberry Pi Zero W" utilizada en medidor *phiNergy* [35].

Según se publica en su página oficial [35], la tarjeta se alimenta a través de un puerto micro USB de $+5.1\ V_{DC}$ y la corriente total depende de los periféricos que estén operando. Al respecto, uno de los principales "sockets" para conectar hardware externo son los pines GPIO (conjunto de 40 contactos agrupados en dos filas en la zona superior de la placa mostrada en *Figura 3.6*) los cuales pueden consumir como máximo $16[mA]$ funcionando individualmente como entradas o salidas.

Para el caso de la *phiNergy*, la *Figura A.3.1* (ver ítem 8.3.1 en Anexos) muestra los pines utilizados por la pinza amperimétrica (sensor AC monofásico), donde se aprecia que su ensamble no supera los cinco contactos. Esto, sin considerar el pin reservado para la tierra de protección. Por otra parte, el tamaño reducido de una *phiNergy* hace que sea fácil de instalar en cualquier tablero eléctrico de

baja tensión, justificándose en gran medida por las dimensiones de la "Raspberry Pi Zero W", cuyas medidas oficiales (en [*mm*]) se ilustran en el esquema de la *Figura A.3.2* del ítem 8.3.1. También, el esquema electrónico oficial se encuentra en la sección Anexos y se exhibe en la *Figura A.3.3*, al final de este trabajo.

Finalmente, la *Figura A.3.4* (ver ítem 8.3.1 en sección Anexos) exhibe los diagramas de bloques más relevantes de las funciones programadas en lenguaje *Python 2*, que son: hacer la lectura desde el sensor AC, construir la base de datos, copiarla y almacenarla en la memoria local (tarjeta SD) y remota (*Server*), y para acceder a la configuración del servicio web (Wi-Fi y variables de tiempo).

3.2.2.1.2 Red de estaciones phiNet

Según publica *Phineal* [23], la plataforma *phiNet* corresponde a: "…una plataforma de información que permite gestionar los datos obtenidos de las estaciones de medición *phiNet 10-11* y de otros dispositivos para procesarlos, guardarlos y generar reportes que serán utilizados en la toma inteligente de decisiones de actividades y procesos, con especial énfasis en las energías renovables y el cuidado del medio ambiente".

Figura 3.7: Estaciones *phiNet 10* y *phiNet 11* emplazadas en Valparaíso para la iniciativa "ValpoSolar" de 2016 [23].

En el caso de *Blockchain GTIME*, cuando se trata de grandes generadores, se utiliza una *vinculación phiNet* entre los datos de los inversores o medidores de facturación dispuestos en planta y los parámetros de radiación solar, temperatura, georreferencia y tiempo medidos simultáneamente por una estación de monitoreo *phiNet10/11* (ver *Figura 3.7*), quien se encargará de subir la información a la base de datos *phiNet*. La idea es utilizar la plataforma *phiNet* para comparar la producción energética de la planta con los patrones de las estaciones de medición ("benchmarking solar"), o bien, para guardar un respaldo de datos de producción de las plantas. Al igual que los medidores *phiNergy*, las estaciones *phiNet 10* y *phiNet 11* se componen de una PCB "Raspberry Pi Zero W", cuyas características técnicas se pueden consultar en el ítem 8.3.1 de la sección Anexos.

Bajo cualquier caso, es posible construir el vector *GTIME* y, cada 20 minutos, establecer la comunicación M2M con el *Server* para anexar un nuevo bloque a la *blockchain*; creando un historial inmutable de las transacciones a lo largo del tiempo.

3.2.2.2 Miners

Tienen el rol de *mineros* de la *Blockchain GTIME* (ver *Figura 3.8*). Son minicomputadores que se encargan de descentralizar la validación de las transacciones y de encontrar el *hash* de los bloques entrantes para agregarlos a la *blockchain*. Respecto de los *Miners* que son propiedad de *Phineal* es bueno comentar que están programados con un "delay" o retraso de tiempo aleatorio para el sellado de bloques, con el propósito de impedir que un único nodo calcule periódicamente el *hash* de estos.

Además, hasta ahora, y de manera similar a la *cadena de bloques* de *Bitcoin*, *Blockchain GTIME* basa su *protocolo de consenso* en la *PoW*, que como se comentó a lo largo del Capítulo 2, obliga a los *Miners* a encontrar un número *nonce* tal que, una vez dentro de la función criptográfica SHA-256, consiga dar con el *hash* del "bloque postulante". Dicho código alfanumérico debe cumplir con la condición de empezar con una cantidad definida de bits con cuantía "0" conforme estipule la *dificultad de minado* del protocolo *GTIME*. **Obs:** Para resolver dudas conceptuales de cómo operan la *minería* y el *protocolo de consenso*, se recomienda que lea el ítem 8.2.1 de la sección Anexos, al final de este documento.

El hecho de que no existan incentivos económicos para hacer *minería* se traduce en que la "competencia" entre *Miners* no tenga cabida. Al respecto, el protocolo *GTIME* establece que minuto a minuto los nodos validadores "pregunten" al *Transaction* si hay algún bloque que "sellar". Ante dicha solicitud, el *Server* reparte entre todos los *mineros* de la red un número aleatorio que servirá posteriormente para seleccionar al *Miner* con el *tiempo de inicio de minado* más bajo.

Figura 3.8: Nodo validador *Miner* [23].

Con ello, cada vez que exista un nuevo bloque "en espera" en el *Transaction*, habrá un único *Miner* en toda la red escogido aleatoriamente para encontrar el *hash*, al menos, en lo que a esta primera versión de validadores de *Phineal* refiere; ya que en caso de ingresar validadores externos a la *Blockchain GTIME*, la competencia será un tema que deberá reevaluarse, conforme se persiga la topología de red de información distribuida tanto para la interacción como para el *minado*.

Este procedimiento de confirmación se traduce, entre otras cosas, en que la potencia demandada sobre la CPU del equipo computacional que se necesita para *minar* el algoritmo criptográfico no sea tan alta, y a su vez, en que el gasto de energía por *hash* del conjunto total de validadores sea equivalente al consumo promedio de un solo *Miner* (cercana a $5[W]$) [26].

En la práctica, un *Miner* está constituido por una tarjeta "Raspberry Pi 3 Modelo B", una placa electrónica que permite la conexión de componentes externos y un interruptor para el encendido/apagado del dispositivo (ubicado en la cara superior del equipo, como se expone en la *Figura 3.8*) [23][26].

En este caso, la tarjeta “Raspberry Pi 3 Modelo B” funciona como minicomputador tanto para calcular los *hashes* que se le asignen, así como para interactuar con las unidades *Transaction* y *Server* del *Blockchain GTIME*; siendo diez veces más rápida que la “Raspberry Pi Zero” [35]. Además, con el fin de que el equipo se conecte automáticamente a internet desde cualquier lugar, es posible acceder a los archivos de la “Raspberry Pi 3 Modelo B” mediante la intervención de la memoria local SD inserta en un “socket” de la PCB para luego configurar la red Wi-Fi deseada. Posterior a ello, todo tipo de comunicación entre el *Miner* y la red distribuida es de tipo M2M.

Al igual que con la tarjeta de la *phiNergy*, el esquema electrónico y las dimensiones milimétricas de la placa PCB utilizada en el *Miner* se encuentran en el ítem 8.3.1 de la sección Anexos, al final de este trabajo (ver *Figura A.3.5* y *A.3.6*, respectivamente). Por su parte, el aspecto de la tarjeta y sus principales componentes, publicados por *Raspberry Pi Foundation* [35], se exponen a continuación:

Figura 3.9: Tarjeta PCB “Raspberry Pi 3 Modelo B” utilizada en validador *Miner* [35].

Características Raspberry Pi 3 Modelo B:

• Chipset Broadcom BCM2837 a 1,2 GHz • Procesador ARM Cortex-A53 de 64 bits y cuatro núcleos • LAN inalámbrica 802.11 b/g/n • Bluetooth 4.1 (posee modo bajo consumo) • Coprocesador multimedia de doble núcleo • Memoria LPDDR2 de 1 GB	• 1 puerto Ethernet 10/100 • 1 conector de vídeo/audio HDMI • 1 conector 3.5mm audio/video compuesto • 4 puertos USB 2.0 • 40 pines GPIO • Conector de pantalla DSI • Ranura de tarjeta microSD

3.2.2.3 Server y Transaction

El *Server*, al igual que cualquier otro servidor, corresponde a la unidad informática encargada de guardar y suministrar los datos requeridos por todos los nodos de la red descentralizada. La información que puede traspasar es diversa, pero para esta aplicación se limita básicamente al tráfico de vectores *GTIME*, a la emisión/recepción de nuevos bloques y al almacenamiento en memoria de la *cadena de bloques*. La unidad *Transaction*, en tanto, es una "tarea" que vive en el *Server* y que está encargada de recibir la data levantada por los nodos medidores, agrupar los vectores *GTIME* en un "bloque abierto", avisar a los *Miners* de que hay una nueva orden de sellado y, finalmente, enviar el bloque sin sellar al validador seleccionado.

Desde una mirada práctica, si bien un servidor no es otra cosa que un computador más robusto en memoria y procesamiento, es bueno comentar que también existen herramientas que llevan la funcionalidad de estos equipos a una versión virtual. Dicha "infraestructura *online*", conocida como *cloud servers*, se obtiene gracias a la existencia de múltiples servidores físicos controlados por un software que se encarga de gestionar su operación y almacenamiento.

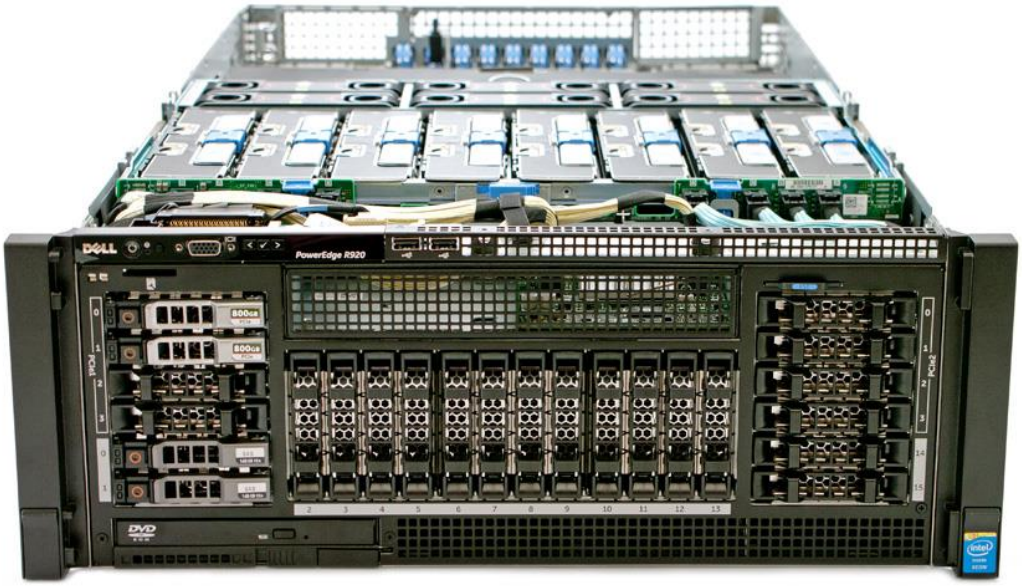


Figura 3.10: Servidor "DELL R920" utilizado por *OpenCloud* para proveer de almacenamiento virtual a *Phineal* [51].

En el caso de *Blockchain GTIME*¸ el *Server* corresponde a uno de los servidores virtuales tipo "VPS SSD" ofrecido por la empresa *OpenCloud*, que cuenta con un disco SSD de $160[GB]$, CPU de 4 núcleos de $2{,}5[GHz]$, $4[TB]$ para tráfico de datos, $8[GB]$ de memoria RAM y una velocidad de enlace a internet de $70[Mbit]$. Todo trabajando, según la web oficial del proveedor [51], en servidores físicos modelo "DELL R920" y cuyo ejemplar se presenta en la *Figura 3.10*.

Desde el punto de vista del software, en coherencia a lo explicado por personal de *Phineal* [23], el sistema operativo sobre el cual está programada la base de datos del *Server*, la *blockchain* y los programas que trabajan en los nodos medidores y validadores es "Raspbian/Linux", que corresponde al software con soporte oficial de la *Raspberry Pi Foundation* bajo la interpretación del sistema operativo *Linux*.

Además, el lenguaje de programación que deben interpretar los equipos *phiNergy*, *red phiNet* y *Miners* para trabajar y comunicarse M2M con el *Server* o cualquier agente de la red descentralizada es "Node.JS", que corresponde a un programa de código abierto de "JavaScript" cuya ventaja principal es la construcción de aplicaciones web escalables y optimizadas [36]. Básicamente, este lenguaje de programación está diseñado para permitir que se lleve a cabo un número elevado de conexiones con el *Server*, sin sacrificar el rendimiento ni la velocidad de las aplicaciones web con las que se trabaje [36]. Ejemplo de ello es que, a nivel de desarrollo de software, las consultas públicas a través de la web, las tareas de *minado* de bloques y las de "request" o actualización de la *blockchain* son asíncronas; vale decir, son operaciones que se pueden ejecutar simultáneamente y sin que se produzca ningún bloqueo en el flujo de trabajo [23].

De este modo, en términos simples, el programa principal con el que se rige el protocolo *GTIME* (*máster*) puede operar de manera más eficiente creando subprocesos (*workers*) que responden de forma paralela a las tareas asignadas, sin que se interrumpa la ejecución del programa (*thread*); siendo válido para las tareas del *Server* y del *Transaction* indistintamente, tal como ilustran los esquemas de la *Figura 3.11* y *Figura 3.12*, respectivamente.

Por último, es bueno comentar que, debido a la cantidad de proyectos adjudicados por la empresa relativos a *blockchain* y energía, el tamaño de la *cadena* almacenada en el *Server* ha pasado a ser un tema relevante. Así, en perspectiva de mejorar la eficiencia del registro de datos y la escalabilidad del sistema de trazabilidad, en el ítem 8.3.2 de la sección Anexos (final de este trabajo), se muestra la apariencia de la *blockchain*, se describen algunos aspectos del proceso de *minería* y se enseña al lector cuál es la tasa de creación de bloques en *Blockchain GTIME* que el equipo de *Phineal* busca optimizar para responder a las nuevas oportunidades de *digitalización* en el sector Energía.

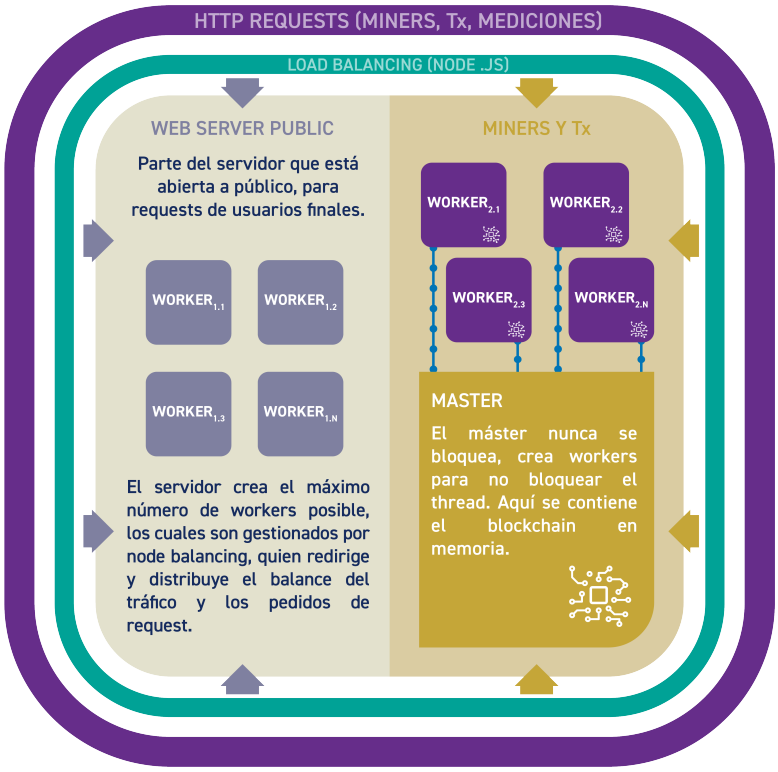


Figura 3.11: Infografía de los subprocesos que utiliza el *Server* para operar de manera más eficiente [23][24].

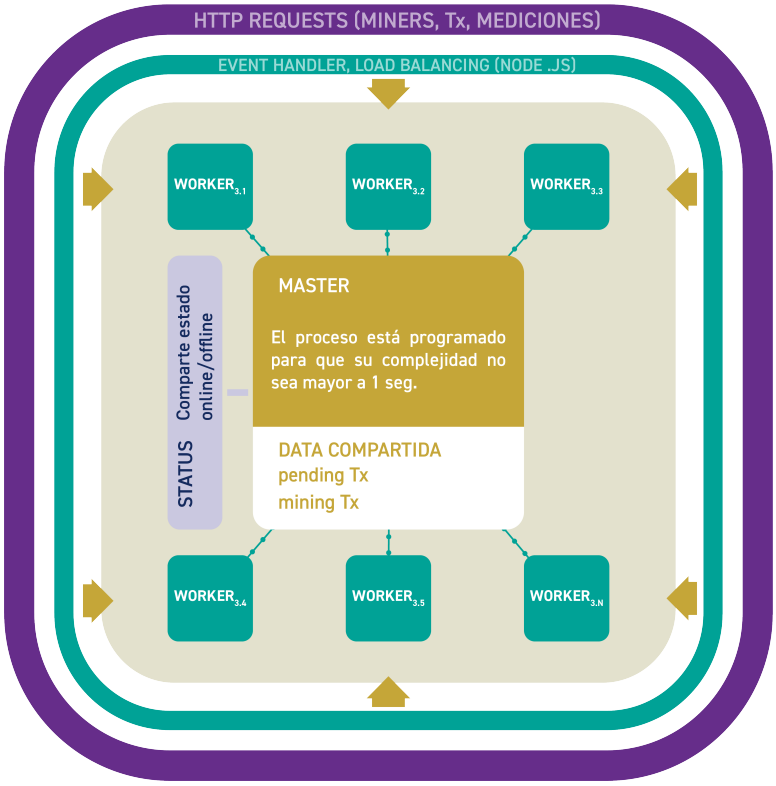


Figura 3.12: Infografía de las consultas y tareas llevadas a cabo en la unidad *Transaction* [23][24].

4. Métodos de asignación de pérdidas en transmisión como herramientas para la trazabilidad de la energía eléctrica

En virtud de la consecución de una *aplicación descentralizada* que logre validar el sistema de trazabilidad perseguido en este trabajo, se estudiaron algunas técnicas de asignación de pérdidas en transmisión para construir una *cadena de suministro* de electricidad. Al respecto, las metodologías incorporadas en la redacción formal de este trabajo fueron las siguientes:

- Método de Bialek [1]
- Método de Kirschen, Allan & Strbac [2]
- Método de Prorrateo o Estampillado Postal [45]

En función de las justificaciones que podrá distinguir el lector en el Capítulo 5, las metodologías seleccionadas en este trabajo fueron el Método de Bialek y el Método de Estampillado Postal, por lo cual las siguientes páginas se ocuparán de exhibir sus principios y procedimientos. El Método de Kirschen, Allan & Strbac, en tanto, se presenta en el ítem 8.4.7 de la sección Anexos, al final de este documento.

Sin desmedro de lo anterior, a nivel general, cabe mencionar que los tres métodos estudiados dependen del estado del sistema, esto es, de las condiciones operacionales al momento de la evaluación (no se basan en criterios incrementales). También es bueno comentar que las tres metodologías entregan una noción del uso parcial o total de la red de transmisión, ya sea mediante la determinación explícita de las líneas utilizadas por los agentes, o a través de la composición de origen de los flujos retirados, o por medio de la composición de reparto de los flujos inyectados, o bien, con nuevos conceptos que llevan implícito el grado de responsabilidad de cada agente en la transferencia de potencia.

En ese sentido, se debe acotar que los dos primeros métodos se hacen cargo de las cualidades espaciales de la red de transporte, vale decir, los resultados están influenciados por la distribución de los flujos a través de las líneas, por los puntos de inyección de potencia y/o por las zonas de retiro de esta. El Método de Prorrateo en cambio, considera al SEP como un sistema energético cerrado desde donde es posible tomar las cantidades totales inyectadas o retiradas del sistema, para posteriormente asignar las pérdidas eléctricas respecto a una proporción de “uso nodal”, haciendo caso omiso de la forma o topología de la red. Es importante tener en cuenta que, bajo cualquiera de los tres casos estudiados, siempre se respeta la *ley de conservación de la energía* y, con ello, la integridad del flujo de carga; por lo que el uso de cualquiera de las metodologías no afecta a la operación del sistema.

Para explicar las metodologías en el orden en que se han presentado, conviene entender que, tanto el Método de Bialek [1] como el Método de Kirschen *et al.* [2], se sustentan en un único supuesto denominado *principio de repartición proporcional* en donde cada flujo de potencia retirado desde un nodo está compuesto en su totalidad por aportes proporcionales de los flujos de potencia inyectados en dicho nodo. Debe notarse que el supuesto exige que se cumplan las *leyes de Kirchhoff*, o dicho de manera más esencial, la *ley de conservación de la carga* y la *ley de conservación de la energía*.

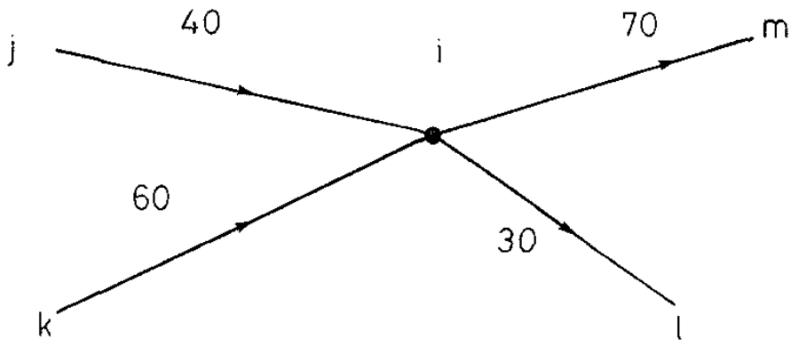


Figura 4.1: Esquema utilizado por J. Bialek para ejemplificar el *principio de repartición proporcional* en la revista *IEEE Proceedings on Generation, Transmition and Distribution* de 1996 [1].

El esquema que antecede es un ejemplo que publicó J. Bialek [1] para explicar el *principio de repartición proporcional*. De la imagen se desprende que las ramas ubicadas a la izquierda del punto de conexión están vertiendo potencia activa o reactiva sobre el nodo *i*, mientras que aquellas ramas de la derecha son las que retiran potencia desde el mismo. En términos cuantitativos, la potencia total a través del nodo en ambos lados de la conexión es de $100[MW]$ (P_{nodal}), de modo que si se denota por P_{i-m}^{j} y P_{i-m}^{k} a las proporciones de potencia que constituyen el flujo P_{i-m} y que provienen desde los terminales *j* y *k*, respectivamente, se puede escribir:

<u>Rama *i-m*:</u>

$$P_{i-m} = P_{i-m}^{j} + P_{i-m}^{k} = \frac{P_{j-i}}{P_{nodal}} \cdot P_{i-m} + \frac{P_{k-i}}{P_{nodal}} \cdot P_{i-m}$$

$$P_{i-m} = \frac{40}{100} \cdot 70[MW] + \frac{60}{100} \cdot 70[MW]$$

$$P_{i-m} = 28[MW] + 42[MW]$$

Análogamente, bajo el mismo supuesto de proporcionalidad, el flujo P_{i-l} puede expresarse en función de las proporciones de los flujos inyectados P_{i-l}^{j} y P_{i-l}^{k} como sigue:

<u>Rama *i-l*:</u>

$$P_{i-l} = P_{i-l}^{j} + P_{i-l}^{k} = \frac{P_{j-i}}{P_{nodal}} \cdot P_{i-l} + \frac{P_{k-i}}{P_{nodal}} \cdot P_{i-l}$$

$$P_{i-l} = \frac{40}{100} \cdot 30[MW] + \frac{60}{100} \cdot 30[MW]$$

$$P_{i-l} = 12[MW] + 18[MW]$$

Además, es posible corroborar que para las ramas de inyección de potencia se concreta el balance. En efecto:

<u>Rama *j-i*:</u>

$$P_{i-m}^{j} + P_{i-l}^{j} = 28[MW] + 12[MW] = 40[MW] = P_{j-i}$$

<u>Rama *k-i*:</u>

$$P_{i-m}^{k} + P_{i-l}^{k} = 42[MW] + 18[MW] = 60[MW] = P_{k-i}$$

Las igualdades anteriores permiten ejemplificar y comprobar que para cada unidad de potencia retirada los aportes proporcionales de inyección son consistentes con el supuesto planteado. En suma, con el uso del *principio de repartición proporcional* en los métodos de Bialek [1] y de Kirschen *et al.* [2], es posible trazar la procedencia, distribución y retiro de los flujos de potencia, consiguiendo una medida implícita del uso parcial de la red de transporte; evitando el análisis particular de caídas de tensión e impedancias para cada línea del sistema y logrando completar la *cadena de suministro*.

El Método de Prorrateo o Estampillado Postal [45], en cambio, supone que los agentes hacen uso de la totalidad de las líneas de transmisión, y con ello, asigna las pérdidas de manera proporcional a los aportes o retiros que los agentes hacen sobre el sistema. A diferencia de las metodologías anteriores, este criterio no permite obtener una *cadena de suministro* completa, por lo que debe apoyarse en otras herramientas o mecanismos para conseguirla; sin que sea una consecuencia exclusiva de esta metodología.

En efecto, el Método de Factores de Distribución elaborado por H. Rudnick y citado en [4] (metodología estudiada pero no incorporada a la redacción formal de esta Memoria), por ejemplo, supone un análisis de sensibilidad para determinar la influencia que tiene un incremento de inyección puntual (o retiro puntual si se trata de cargas) en las variaciones de flujo que experimenta cada línea del sistema de transporte. A

partir de ello, es posible conocer cuáles son las líneas más utilizadas por los agentes (generadores o cargas), especificando la contribución sobre el flujo total y la cantidad de pérdidas producidas. Aún así, este método tampoco arroja una *cadena de suministro* dentro de los resultados, por lo que, a pesar del nivel de detalle en la composición de los flujos, también debe apoyarse en otro tipo de técnicas.

En lo que sigue, se presentarán las metodologías estudiadas con el propósito de evidenciar cuáles son los procedimientos más ventajosos para implementarse en la *aplicación descentralizada* detallada en el Capítulo 5 del presente trabajo.

4.1 Método de Bialek

En su artículo *Tracing the flow of electricity* publicado en 1996, J. Bialek [1] presentó un nuevo método para efectuar el rastreo de flujos de potencia activa o reactiva a través de una red de transmisión, permitiendo conocer al menos uno de los siguientes puntos:

- Contribución de energía inyectada por cada generador para constituir el flujo de potencia que circula por una línea o por una carga en particular.

- Contribución de energía retirada por cada carga para conformar el flujo de potencia que transita por una línea o por un generador puntual.

Para aplicar el Método de Bialek se debe contar previamente con el flujo de carga en estado permanente del SEP a analizar, esto es, estar en conocimiento de los niveles de tensión y ángulos de todas las barras del sistema; y por extensión, contar con los flujos de potencia inyectados por los generadores, los flujos de potencia que circulan a través de las líneas de transmisión, los flujos de potencia retirados por las cargas y las pérdidas eléctricas asociadas al proceso completo.

Considerando lo anterior, el autor [1] presenta dos algoritmos diferentes para la resolución del problema de trazabilidad: *algoritmo de inyecciones de potencia* y *algoritmo de retiros de potencia*. Dichos conjuntos de instrucciones trabajan con flujos de potencia sin pérdidas en las líneas, es decir, se asume que los flujos de potencia en ambos extremos de una línea son iguales; por lo que, para desplazar las pérdidas totales incurridas en la red, se deben realizar ajustes sobre las inyecciones de los generadores y/o retiros de las cargas del SEP.

La condición de los algoritmos (de que los flujos de potencia que atraviesan las líneas no incorporen pérdidas) se puede alcanzar mediante tres alternativas distintas: *flujos gruesos*, *flujos medios* y *flujos netos*. Dependiendo de la forma de resolución que se escoja, se culmina el problema asignando las pérdidas eléctricas totales a uno o a ambos participantes de la red, a saber: *demandas*, *generadores*, o bien, *demandas y generadores*, según sea el caso.

En lo que sigue, se presentan los algoritmos y posteriormente las distintas formas de resolución propuestas por J. Bialek para la determinación de las contribuciones energéticas que cada generador o demanda tiene sobre los flujos de potencia del sistema.

4.1.1 Algoritmo de inyecciones de potencia

Si para un SEP de n barras la expresión para la potencia total a través de un nodo i se define a partir de las inyecciones de potencia generada P_{G_i} en dicho punto, se puede escribir:

$$P_i = \sum_{j \in \alpha_i^{(u)}} \left|P_{i-j}\right| + P_{G_i} \quad ; \quad \forall\, i \in \{1, n\}$$

Donde:

$\alpha_i^{(u)}$: corresponde al conjunto de nodos o barras que surten potencia directamente al nodo i.

P_{i-j}: es el flujo de potencia desde el terminal j al terminal i, bajo el supuesto de ausencia de pérdidas.

P_{G_i}: representa la potencia inyectada por un generador en la barra i.

Entonces, asumiendo que no hay pérdidas, se puede afirmar que $\left|P_{i-j}\right| \equiv \left|P_{j-i}\right|$ y con ello es posible relacionar las variables de la barra i con la potencia nodal de la barra j a través de un coeficiente de proporción definido por:

$$c_{ji} = \frac{\left|P_{j-i}\right|}{P_j} = \frac{\left|P_{i-j}\right|}{P_j}$$

De modo que es posible reescribir la expresión de P_i como sigue:

$$P_i = \sum_{j \in \alpha_i^{(u)}} c_{ji} \cdot P_j + P_{G_i} \quad ; \quad \forall\, i \in \{1, n\}$$

O equivalentemente:

$$P_i - \sum_{j \in \alpha_i^{(u)}} c_{ji} \cdot P_j = P_{G_i} \quad ; \quad \forall\, i \in \{1, n\}$$

Luego, si se interpreta la igualdad anterior en términos matriciales, es posible expresar:

$$\langle A_u \rangle \cdot \langle P_{nodal} \rangle = \langle P_G \rangle$$

Donde:

$\langle A_u \rangle$: representa la matriz de distribución de inyecciones de potencia, cuya dimensión es $n \times n$.

$\langle P_{nodal} \rangle$: es el vector que agrupa la potencia nodal total de cada barra del sistema.

$\langle P_G \rangle$: es el vector que agrupa la potencia inyectada en los nodos de generación del sistema.

Es importante explicitar que los elementos de la matriz $\langle A_u \rangle$ vienen definidos por:

$$A_{u_{ij}} = \begin{cases} 1 & ; \quad i = j \\ -c_{ji} = -\dfrac{|P_{j-i}|}{P_j} & ; \quad j \in \alpha_i^{(u)} \\ 0 & ; \quad e.o.c \end{cases}$$

Con lo anterior, se evidencia que el elemento matricial $A_{u_{ij}}$ será igual a $-c_{ji}$ únicamente si el flujo de potencia circula desde el terminal *j* al terminal *i*.

En consecuencia, si $\langle A_u \rangle$ admite inversa, entonces es posible despejar el vector $\langle P_{nodal} \rangle$ del sistema matricial de ecuaciones, según:

$$\langle P_{nodal} \rangle = \langle A_u \rangle^{-1} \cdot \langle P_G \rangle$$

Donde cada elemento del vector de potencias nodales netas del sistema viene expresado por:

$$P_i = \sum_{k=1}^{n} A_{u_{ik}}^{-1} \cdot P_{G_k} \quad ; \quad \forall\, i \in \{1, n\}$$

Aquí se debe notar que esta última igualdad exhibe el aporte que cada generador *k* del SEP hace sobre la potencia nodal de la barra *i*.

Por su parte, en virtud del *principio de repartición proporcional*, si el flujo de potencia que circula por la línea *i-l* del sistema de transmisión experimenta un retiro de potencia por parte del nodo *i*, se puede escribir:

$$|P_{i-l}| = \frac{|P_{i-l}|}{P_i} \cdot P_i = \frac{|P_{i-l}|}{P_i} \cdot \sum_{k=1}^{n} A_{\mathcal{U}ik}^{-1} \cdot P_{G_k} \quad ; \quad \forall\, i \in \{1, n\}$$

O equivalentemente:

$$|P_{i-l}| = \sum_{k=1}^{n} \frac{|P_{i-l}|}{P_i} \cdot A_{\mathcal{U}ik}^{-1} \cdot P_{G_k} = \sum_{k=1}^{n} D_{i-l\,,k}^{G} \cdot P_{G_k} \quad ; \quad \forall\, l \in \alpha_i^{(d)}$$

Donde:

$D_{i-l\,,k}^{G}$: *factor de distribución de generación topológico*, cuantifica la proporción de potencia que el generador *k* aporta sobre el flujo de potencia que circula por la línea *i-l*.

$\alpha_i^{(d)}$: corresponde al conjunto de nodos o barras alimentadas directamente por el nodo *i*.

Análogamente para los consumos del SEP, en función del *principio de repartición proporcional*, se puede calcular la contribución de potencia de cada generador *k* en el flujo demandado por una carga *i* de acuerdo la siguiente expresión:

$$P_{L_i} = \frac{P_{L_i}}{P_i} \cdot P_i = \frac{P_{L_i}}{P_i} \cdot \sum_{k=1}^{n} A_{\mathcal{U}ik}^{-1} \cdot P_{G_k} \quad ; \quad \forall\, i \in \{1, n\}$$

La igualdad anterior explicita la posibilidad de conocer la procedencia del conjunto de potencias que alimenta a una determinada carga y, para un determinado intervalo de tiempo, dar respuesta al problema de trazabilidad de energía eléctrica.

4.1.2 Algoritmo de retiros de potencia

Si ahora la expresión para la potencia total a través de una barra *i* que pertenece a un SEP de n barras se describe en función de la potencia demandada $P_{\mathcal{D}_i}$ en dicho punto, es posible escribir:

$$P_i = \sum_{l \in \alpha_i^{(d)}} |P_{i-l}| + P_{\mathcal{D}_i} \quad ; \quad \forall\, i \in \{1, n\}$$

Donde:

$\alpha_i^{(d)}$: corresponde al conjunto de nodos o barras que surten potencia directamente al nodo *i*.

P_{i-l}: es el flujo de potencia desde el terminal *l* al terminal *i*, bajo el supuesto de ausencia de pérdidas.

$P_{\mathcal{D}_i}$: representa la potencia retirada por una carga en el nodo *i*.

Al igual que en el algoritmo anterior, bajo el supuesto de que no hay pérdidas eléctricas, se define un coeficiente de proporción c_{li} para vincular a las variables de la barra *i* con el valor de P_l según:

$$c_{li} = \frac{|P_{l-i}|}{P_l} = \frac{|P_{i-l}|}{P_l}$$

Lo anterior permite reescribir la expresión para la potencia nodal en la barra *i* como sigue:

$$P_i = \sum_{l \in \alpha_i^{(d)}} c_{li} \cdot P_l + P_{\mathcal{D}_i} \Leftrightarrow P_i - \sum_{l \in \alpha_i^{(d)}} c_{li} \cdot P_l = P_{\mathcal{D}_i} \quad ; \quad \forall\, i \in \{1, n\}$$

Se debe notar que la expresión anterior describe un sistema matricial que puede agruparse como:

$$\langle A_d \rangle \cdot \langle P_{nodal} \rangle = \langle P_{\mathcal{D}} \rangle$$

Donde:

$\langle A_d \rangle$: representa la matriz de distribución de retiros de potencia, cuya dimensión es $n \times n$.

$\langle P_{nodal} \rangle$: corresponde al vector que agrupa la potencia nodal total de cada barra del sistema.

$\langle P_{\mathcal{D}} \rangle$: describe al vector que agrupa la potencia retirada en los nodos de demanda del sistema.

Adicionalmente, el autor señala que los elementos de la matriz $\langle A_d \rangle$ están definidos por la relación:

$$A_{d_{il}} = \begin{cases} 1 & ; \quad i = j \\ -c_{li} = -\dfrac{|P_{l-i}|}{P_l} & ; \quad l \in \alpha_i^{(d)} \\ 0 & ; \quad e.o.c \end{cases}$$

Notar que esta última expresión establece que el terminal *l* debe ser una barra alimentada por el terminal *i* para que el elemento matricial $A_{d_{il}}$ sea distinto de 0 y 1.

Luego, si el determinante de la matriz $\langle A_d \rangle$ es distinto de cero, entonces es posible despejar el vector de potencias nodales, como sigue:

$$\langle P_{nodal} \rangle = \langle A_d \rangle^{-1} \cdot \langle P_{\mathcal{D}} \rangle$$

Donde cada elemento del vector despejado está descrito por la igualdad:

$$P_i = \sum_{k=1}^{n} A_{d_{ik}}^{-1} \cdot P_{\mathcal{D}_k} \quad ; \quad \forall\, i \in \{1, n\}$$

Esta última ecuación exhibe la solicitud de potencia que cada demanda *k* del SEP hace sobre la potencia nodal de la barra *i*.

De forma similar al algoritmo descrito en el ítem 4.1.1, el *principio de repartición proporcional* ofrece la posibilidad de calcular los aportes de potencia que hace la línea *i-j* sobre el flujo inyectado en cada barra *i* del SEP. En efecto:

$$\left|P_{i-j}\right| = \frac{\left|P_{i-j}\right|}{P_i} \cdot P_i = \frac{\left|P_{i-j}\right|}{P_i} \cdot \sum_{k=1}^{n} A_{d_{ik}}^{-1} \cdot P_{\mathcal{D}_k} \quad ; \quad \forall\, i \in \{1, n\}$$

O bien descrito por:

$$\left|P_{i-j}\right| = \sum_{k=1}^{n} \frac{\left|P_{i-j}\right|}{P_i} \cdot A_{d_{ik}}^{-1} \cdot P_{\mathcal{D}_k} = \sum_{k=1}^{n} D_{i-j\,,k}^{\mathcal{D}} \cdot P_{\mathcal{D}_k} \quad ; \quad \forall\, j \in \alpha_i^{(u)}$$

Donde:

$D_{i-j\,,k}^{\mathcal{D}}$: *factor de distribución de carga topológico*, cuantifica la proporción de potencia que la carga *k* demanda desde el flujo de potencia que circula por la línea *i-j*.

$\alpha_i^{(u)}$: corresponde al set de barras alimentadas directamente por el nodo *i*.

También es posible conocer el requerimiento de potencia que cada demanda *k* tiene sobre el flujo inyectado por un generador *i*. En conformidad con el *principio de repartición proporcional* se consigue:

$$P_{G_i} = \frac{P_{G_i}}{P_i} \cdot P_i = \frac{P_{G_i}}{P_i} \cdot \sum_{k=1}^{n} A_{d_{ik}}^{-1} \cdot P_{\mathcal{D}_k} \quad ; \quad \forall\, i \in \{1, n\}$$

Esta última igualdad da la opción de conocer la “ruta” o “camino” que sigue la potencia suministrada por un generador para alimentar a un parque de cargas. Además, se puede inferir que si se emplea este algoritmo a través del tiempo, hay cabida para la resolución del problema de trazabilidad de la energía eléctrica.

4.1.3 Rastreo de electricidad según asignación de pérdidas

Dependiendo del agente del SEP al que se le atribuyan las pérdidas totales de la transferencia energética, el autor [1] propone tres formas distintas para representar los flujos de potencia a través de la red de transporte, esto es, a través de: *flujos brutos (o gruesos)*, *flujos medios* y *flujos netos*. El objetivo y descripción general de cada resolución se exponen a continuación:

- **Trazabilidad eléctrica usando Flujos Gruesos:**
  La idea es atribuir las pérdidas eléctricas al bloque de demanda del sistema. En términos cuantitativos, lo anterior se consigue utilizando el *algoritmo de inyecciones de potencia* para agregar el “consumo extra” a las cargas del SEP. Para cumplir con el supuesto de ausencia de pérdidas en la red de transporte, se modifican las demandas nodales del sistema, mientras que la generación nodal queda intacta.

- **Trazabilidad eléctrica usando Flujos Medios:**
  El propósito es responsabilizar tanto al bloque generador como al bloque consumidor de las pérdidas eléctricas provocadas por el uso de la red de transmisión. En términos cuantitativos, lo anterior se modela con el uso del *algoritmo de inyecciones de potencia* o bien utilizando el *algoritmo de retiros de potencia*. En este caso, para cumplir con el supuesto de que no hay pérdidas en la red de transporte, se deben modificar las potencias nodales de los generadores y de las cargas del sistema.

- **Trazabilidad eléctrica usando Flujos Netos:**
  La intención es cargar las pérdidas eléctricas al bloque de generación del SEP. En términos cuantitativos, lo anterior se consigue empleando el *algoritmo de retiros de potencia* para aminorar el suministro de los generadores del sistema. Para cumplir con el supuesto de ausencia de pérdidas en transmisión, se modifica la producción nodal, mientras que el consumo nodal del sistema queda intacto.

Para efectos de una mejor comprensión de la metodología, se invita al lector a revisar la sección Anexos en los ítems 8.4.1, 8.4.3 y 8.4.5, donde se presenta un SEP de prueba sobre el cual se aplican las tres formas de resolución antes descritas y se detalla la obtención de los resultados. El sistema eléctrico empleado es el denominado "Modelo de Garver" y ha sido citado por M. Chamba [3] para explicar el uso del Método de Bialek en las liquidaciones de las transacciones comerciales de dicho SEP, por lo cual los "datos de entrada" se han extraído desde su trabajo. Con el desarrollo expuesto en los ítems mencionados, se pretende enseñar cómo se conforman los flujos de potencia a partir de las contribuciones de los agentes, además de discriminar al o los responsables de las pérdidas eléctricas del sistema. Adicionalmente, en los ítems 8.4.2, 8.4.4 y 8.4.6 de la sección Anexos, se muestran los códigos empleados en MATLAB para la resolución del Método de Bialek con Flujos Gruesos, Flujos Medios y Flujos Netos, respectivamente.

4.2 Método de Prorrateo o Estampillado Postal

Si bien este método ha sido ampliamente utilizado como mecanismo de tarificación de la transmisión [45], en este trabajo se aplicará una adaptación de la fórmula sobre variables físicas. En ese sentido, tal como se mencionó al comenzar este capítulo, el Método de Prorrateo o Estampillado Postal considera al sistema eléctrico como "una gran bolsa de energía", haciendo caso omiso de las propiedades espaciales de la red: como la ubicación del parque generador o del grupo de cargas. Desde un punto de vista global, en conformidad a la *ley de conservación de la energía*, lo anterior implica que la asignación de pérdidas para cada agente del SEP actúa bajo el supuesto de que estos utilizan la totalidad del sistema de transmisión y, por tanto, la repartición debe ejecutarse en función de las cantidades totales suministradas y retiradas del sistema.

Esto último implica que no existe intención alguna de determinar cuáles son las inyecciones de potencia que conforman el flujo que circula por una línea o que consume una carga particular, pero sí permite asignar un "cargo nodal" en relación con las pérdidas totales de las transferencias efectuadas. Si bien, tradicionalmente este método se ha utilizado para repartir las pérdidas entre las cargas [45], igualmente puede ocuparse para distribuir el costo físico de transporte entre los generadores, o bien, a través de una versión mixta. Con todo lo anterior, se puede inferir que esta metodología responde parcialmente al problema de trazabilidad, por cuanto responsabiliza a la demanda por los retiros de potencia y por el *Efecto Joule* asociado a su etapa de transporte, pero al mismo tiempo desconoce los puntos de suministro

y la distribución de los flujos por las líneas, con lo cual es indiferente el lugar de la red en el que se producen las pérdidas.

Note que, conforme a lo que explican los autores de [45], lo anterior puede generar enormes distorsiones económicas cuando se valoriza el costo físico de transporte. Por ejemplo, si una carga que está lejos de un generador y que retira igual magnitud de potencia que otra carga conectada directamente a la barra de dicho generador, el método determina que ambas cargas hacen el "mismo" uso de la red de transmisión y, por ende, se adjudican igual cantidad de pérdidas. De aquí se infiere que el problema es más conflictivo en sistemas de gran extensión radial donde los centros de producción se alejan considerablemente de los centros de consumo; pero puede ser ventajoso en sistemas más enmallados o en anillo, donde exista una distribución más homogénea entre cargas y generadores. Naturalmente que la longitud de las líneas no es la única variable en juego, siendo especialmente relevantes los parámetros eléctricos de los conductores, la cargabilidad del transporte y las técnicas de compensación reactiva con las que cuente la red.

En términos prácticos, la aplicación del Método de Prorrateo consta de dividir las pérdidas totales de transmisión para repartirlas según la cantidad de flujo retirado por las cargas del sistema. Es evidente que el procedimiento exhibe su mayor ventaja en la sencillez de cálculo, donde la expresión general viene definida por:

$$P_{losses_i} = \frac{\mathcal{D}_i}{\mathcal{D}_{total}} \cdot P_{losses_{total}}$$

Donde:

P_{losses_i}: asignación de pérdidas eléctricas a la demanda de la *barra i* del sistema

$\mathcal{D}_i$: demanda conectada en la *barra i*

$\mathcal{D}_{total}$: demanda total del sistema

$P_{losses_{total}}$: pérdidas eléctricas totales del sistema

Por último, en el ítem 8.4.10 de la sección Anexos (al final de este documento) se exhibe un SEP de 6 barras y 9 líneas utilizado para ejemplificar el Método de Kirschen, Allan & Strbac en el ítem 8.4.8, pero ahora se somete al Método de Estampillado Postal. Con esto, se pretende mostrar cómo se reparten las pérdidas entre demandas, de manera independiente a la topología de la red y a la ubicación de los puntos de inyección.

4.3 Cuadro comparativo de las metodologías estudiadas

El siguiente cuadro resumen pretende enseñar las características más importantes de los métodos de asignación de pérdidas estudiados en este capítulo. La *Tabla 4.1*, además, servirá en el Capítulo 5 como base para dilucidar cuáles son los procedimientos más idóneos para construir una *cadena de suministro* instantánea consistente con los objetivos propuestos en este documento. Además, con el fin de tener un panorama más amplio de los métodos aplicables al problema, también se agregó el Método de Factores de Distribución, aun cuando este no fue analizado formalmente en este trabajo.

Método	**Referencia**	**Supuesto**	**Flujo**	**Agente responsable de las pérdidas**	**Cadena de suministro resultante**
Bialek - Flujos Gruesos	[1]	Principio de repartición proporcional	AC & DC	Demanda	Completa
Bialek - Flujos Medios	[1]	Principio de repartición proporcional	AC & DC	Generación & Demanda	Completa
Bialek - Flujos Netos	[1]	Principio de repartición proporcional	AC & DC	Generación	Completa
Kirschen, Allan & Strbac	[2]	Principio de repartición proporcional	AC & DC	Generación	Completa
Estampillado Postal	[45]	Uso total de la red	No aplica	Generación o Demanda	Generación + pérdidas o Demanda + pérdidas
Factores de Distribución*	Citado en [4]	Uso parcial de la red	AC & DC	Generación o Demanda	Generación + pérdidas o Demanda + pérdidas

Tabla 4.1: Principales características de las metodologías estudiadas en el Capítulo 4. Elaboración propia.

5. Aplicación descentralizada, programación de Smart Contract y validación del sistema de trazabilidad de energía eléctrica bajo el protocolo GTIME

En vista de que ya se han enseñado los principales conceptos y definiciones de *blockchain* y *smart contracts* en el Capítulo 2, y dado que en el Capítulo 3 se ha presentado el sistema y plataforma que operará como "base de datos distribuida" de las inyecciones y retiros de la transferencia energética; en este capítulo se mostrará el desarrollo de la *aplicación descentralizada* que permite validar el sistema de trazabilidad con *blockchain* propuesto, mediante la obtención de una *cadena de suministro* instantánea de electricidad entre PV Salvador y el cliente final en Santiago.

Para cumplir con el objetivo general de este trabajo, se seleccionará uno o más de los métodos de asignación de pérdidas estudiados en el Capítulo 4 para incorporarlos a una *aplicación descentralizada* que, mediante un *smart contract*, logre gestionar los datos empíricos de producción, consumo y de flujos circulantes entre los agentes de la transacción.

La elección de las metodologías con que se distribuirán las pérdidas, la codificación de las restricciones que modelan el comportamiento físico del problema y la construcción del *smart contract* se llevarán a cabo en los ítems que siguen, exhibiendo las fortalezas y desafíos de este nuevo sistema de trazabilidad de la energía.

5.1 Validación de la trazabilidad de flujos entre producción y consumo

Este primer ítem tiene por finalidad formular la propuesta metodológica de asignación de pérdidas a utilizar en la *aplicación descentralizada* que se implementará en conjunto con la empresa *Phineal* y que usará *blockchain* como herramienta tecnológica del sistema de trazabilidad.

Para discriminar las metodologías que mejor se adaptan a la realidad nacional, debe considerarse la actual Ley de Transmisión (Ley 20.936) [37], que en el artículo 115° establece que el pago por uso de los sistemas de transmisión nacional, zonal y dedicada es íntegramente asumido por los clientes finales libres y sometidos a regulación de precios. De este modo, desde la óptica del problema de trazabilidad y en coherencia a la legislación vigente, lo más prudente es adjudicar el costo físico de la transferencia energética a los clientes finales y, por tanto, la solución debe contemplar el desplazamiento de las pérdidas hacia las cargas o centros de consumo del SEP. Advierta que, según los alcances de este trabajo, sólo se repartirán las pérdidas incurridas a nivel nacional, excluyendo del problema el fenómeno a nivel zonal y distribución.

Dicho esto, y de acuerdo con el cuadro comparativo presentado en el ítem 4.3 del capítulo anterior, así como con el análisis de los ejemplos revisados en la sección Anexos (ítem 8.4 al final de este trabajo), las metodologías más idóneas en lo que a términos físicos refiere son:

- Método de Bialek usando Flujos Gruesos
- Método de Prorrateo o Estampillado Postal

Se debe acentuar que para seleccionar ambos métodos hay diversas razones. Por un lado, si se pone atención a la *Tabla 4.1*, podrá apreciar que uno de los puntos fuertes del procedimiento de Flujos Gruesos de Bialek es que permite obtener una *cadena de suministro* completa a nivel sistémico, esto es, consigue relacionar las contribuciones de inyección de todos los generadores de un SEP y los flujos que circulan por la red, para mostrar la repartición de pérdidas y la composición de origen de los flujos retirados por cada carga del sistema. Sin embargo, note que la aplicación ortodoxa de este mecanismo implicaría un enorme desafío en términos de implementación, puesto que, para efectuar la trazabilidad de flujos entre la planta PV Salvador y el cliente final en Santiago de manera empírica, sería necesario acceder en tiempo real a todos los equipos de telemetría que el Coordinador Eléctrico Nacional tiene dispuestos a nivel de transmisión nacional y zonal, además de toda la información resultante del flujo de carga instantáneo a nivel de distribución. Se trata de una tarea titánica.

Por otra parte, la repartición proporcional de pérdidas a través del Método de Estampillado Postal tiene la virtud de ser un procedimiento extremadamente sencillo y evidente, lo que le reporta grandes ventajas en términos de implementación, puesto que sólo se necesita acceder a los datos instantáneos de generación global, demanda global y retiro puntual. No obstante, en coherencia a la *Tabla 4.1*, este mecanismo por sí solo no puede construir una *cadena de suministro* y su ejecución a nivel sistémico permea en distorsiones económicas importantes en lo que a atribución de uso de las líneas concierne, sobre todo en redes de gran extensión radial y con una distribución espacial heterogénea de productores y consumidores; como ocurre, por ejemplo, a nivel de transmisión nacional.

En suma, con fin de adoptar una propuesta metodológica que fuera factible de incorporar en los códigos de la *aplicación descentralizada*, se decidió combinar sinérgicamente el Método de Bialek y el de Prorrateo aplicándolos sobre un modelo simplificado del Sistema Eléctrico Nacional (SEN), sin contemplar las pérdidas a nivel zonal ni en distribución. De este modo, con la cantidad acotada de información empírica que supone un “SEN reducido”, es posible ejecutar más cómodamente el procedimiento de Flujos Gruesos de Bialek a nivel de transmisión nacional para obtener una *cadena de suministro* en la que se exhiba tanto el *pool* de proveedores energéticos como las pérdidas atribuibles a la barra de abastecimiento de la ciudad

de Santiago. Luego, en conocimiento de dichas pérdidas locales, se puede desplazar el costo físico de transporte al cliente final conforme al Método de Estampillado Postal. Esto último implica que, a nivel de transmisión zonal y de distribución, se ha considerado una repartición espacial más homogénea de las cargas respecto del punto de inyección (barra alimentadora en Santiago), lo que parece ser más razonable que hacer la misma suposición a nivel de transmisión nacional.

Además, dado que el objetivo principal es validar el sistema de trazabilidad mediante la elaboración de una *cadena de suministro* empírica y en tiempo real, se decidió utilizar los datos de producción de energía de PV Salvador medidos y validados por la plataforma de certificación solar *Sello Sol* (basada en *Blockchain GTIME*), al igual que con los datos de consumo del cliente final. De este modo, las inyecciones y los retiros que formarán parte de la *cadena de suministro* construida con la *aplicación descentralizada* contarán con todos los atributos que el protocolo *GTIME* puede conceder.

Por su parte, dadas las dificultades para conseguir que las pérdidas adquieran una cualidad empírica de forma instantánea, se optó por efectuar flujos de carga horarios sobre el modelo simplificado del SEN a proyectar, en donde los datos de entrada del problema corresponden a mediciones horarias registradas en el portal de información pública del Coordinador Eléctrico Nacional [38]. Para ello, se escogió un día arbitrario del año y los estados horarios se definieron según la curva de producción de PV Salvador en dicho día. Así, se pretende simular que se dispone de información instantánea para aplicar las metodologías seleccionadas y, con ello, se posibilita el cálculo "en tiempo real" de las pérdidas que formarán parte de la *cadena de suministro*, al menos, para efectos de este prototipo.

5.1.1 Modelo simplificado del Sistema Eléctrico Nacional

Con el propósito de estudiar la interacción energética entre dos puntos conectados al SEN, y en función de la propuesta metodológica que se utilizará a nivel de transmisión nacional, zonal y de distribución, el análisis eléctrico estará enfocado en un SEP sometido a un flujo de carga en estado permanente que muestre la distribución de los flujos de potencia para distintos escenarios. Conviene que el lector comprenda que, como se anticipó en el ítem anterior, la idea es utilizar los Flujos Gruesos del Método de Bialek con el fin de repartir las pérdidas totales sobre cada barra de demanda en un modelo reducido del SEN, para luego usar el Método de Prorrateo o Estampillado Postal sobre la barra alimentadora de Santiago y, con ello, asignar las pérdidas al cliente final según la magnitud de su consumo.

En vista de lo anterior, se trabajó en reducir un segmento del circuito oficial de la red de transmisión nacional del SEN (divulgado por el Coordinador Eléctrico Nacional) [38], que incluye tanto a los agentes de la transacción energética como a las líneas y barras más importantes conectadas a ellos. Dado que lo más relevante son las cuantías instantáneas de las potencias activas que transitan por dicha porción del SEN (en función de los objetivos de este trabajo), se decidió utilizar los siguientes criterios:

- Se realizará un modelo simplificado basado en un estado único del sistema, aun cuando se sabe que un SEP está mutando constantemente. Así, sobre dicho "SEN reducido", se ejecutarán los flujos de carga horarios pertinentes para la obtención de pérdidas locales en la barra del cliente final.

- El emisor de las inyecciones será la planta PV Salvador emplazada en El Salvador, Atacama.

- El receptor de dichas inyecciones será la oficina de *Phineal* situada en Providencia, Santiago.

- El segmento del SEN a reducir deberá contener a la barra Diego de Almagro (conexión PV Salvador) y la barra Cerro Navia (conexión habitual de un cliente final en la ciudad de Santiago).

- Conforme a las instalaciones actuales del SEN publicadas por el Coordinador [38], la tensión nominal predominante a lo largo del sistema de transmisión nacional deberá ser de $220[kV]$.

- Para el estado seleccionado, se sumarán todas las inyecciones por barra para crear generadores equivalentes por barra; haciendo lo propio con los retiros, es decir, estableciendo cargas equivalentes por barra. La única excepción será el generador PV Salvador, a quien se le dará prioridad debido al análisis que persigue este documento.

- Todas las potencias generadas y consumidas deberán obedecer a los criterios de operación real del SEN, esto es, serán magnitudes físicas que satisfacen plenamente las condiciones técnicas y económicas del sistema para un instante determinado (piénsese en despacho óptimo y operación económica del parque generador, niveles adecuados de cargabilidad en las líneas, niveles aceptables de tensión en las barras, etc.).

- Dado el punto anterior y en sintonía con los objetivos de este documento, sólo se utilizarán líneas para construir el SEN reducido, dejando fuera del modelo a todo tipo de transformadores y a cualquier elemento de compensación reactiva.

Con las condiciones explicitadas anteriormente, se optó por precisar las 14:00 hrs. del 28 de febrero de 2018 como instante de definición de estado del sistema, con el fin de "privilegiar" la producción fotovoltaica del agente generador y debido a la abundancia de información publicada y disponible. Para asegurar que los flujos de potencia responden fielmente a la operación real del SEN, se recurrió a la plataforma de información pública del Coordinador Eléctrico Nacional [38] para descargar los datos de

“generación real” y “demanda real” del sistema a la hora y fecha seleccionadas. Posteriormente se hizo un listado en Excel de los generadores y cargas más relevantes, permitiendo la priorización de las líneas más utilizadas en dicho momento (material complementario entregado junto a este trabajo). Asimismo, debido a la ubicación de los agentes involucrados en la transferencia energética, se decidió acotar la simplificación del SEN al centro del circuito; esto es, comenzando en la barra Paposo (Taltal, Región de Antofagasta) y culminando en la barra Ancoa (Colbún, Región del Maule), excluyendo por completo los segmentos del sistema de transmisión nacional que se conectan en $500[kV]$ con la zona central del país: como el sistema norte en $220[kV]$ que recorre la Región de Arica y Parinacota, Región de Tarapacá y la mayor parte de la Región de Antofagasta; como también el sistema sur en $220[kV]$ que comienza en la Región del Biobío y culmina en la Región de Los Lagos.

Respecto de la recolección de datos, la *Tabla 5.1* muestra los retiros de potencia más importantes efectuados el 28 de febrero de 2018 a las 14:00 hrs. en la porción escogida del SEN. La *Tabla 5.2*, en tanto, expone las inyecciones más importantes realizadas sobre el sistema en la fecha y hora convenidas. En ambas tablas se expone la cuantía del flujo de potencia y la barra a la cual están conectados los agentes del SEP, además de una clasificación cualitativa en relación con el tipo de cliente o tipo de central, respectivamente.

N°	Demandas seleccionadas SEN	P_{ret} [MW]	Tipo	Barra de conexión	$P_{ret\ eq.}$ [MW]
1	Minera Candelaria	92,13	Industrial	Cardones	168,39
2	EMELAT	76,27	Industrial	Cardones	
3	Chilquinta	395,62	Distribución	Quillota	812,60
4	CONAFE	275,89	Distribución	Quillota	
5	Minera Los Pelambres	141,10	Industrial	Quillota	
6	Angloamerican Sur Los Bronces	179,25	Industrial	Polpaico	179,25
7	ENEL Distribución	1979,16	Distribución	Cerro Navia	1979,16
8	CGE Distribución	1148,52	Distribución	Alto Jahuel	1368,05
9	CODELCO El Teniente	219,53	Industrial	Alto Jahuel	
Total P_{ret} [MW]		4507,45			

Tabla 5.1: Selección de cargas más relevantes del sistema para el 28/02/2018 a las 14:00 hrs. según el criterio de “demanda horaria por cliente” divulgado en el portal de información pública del Coordinador Eléctrico Nacional [38].

La simplificación del SEN también significó, conforme se expone en la *Tabla 5.1* y *Tabla 5.2*, establecer elementos equivalentes por barra de modo tal de atribuir el total de flujos inyectados a un único generador equivalente, o bien, el total de flujos retirados a una única carga equivalente. Toda información relativa a

generación y demanda horarias se puede consultar en las planillas Excel del archivo “Neteo Energético horario febrero 2018.xlsx” que contiene la información descargada desde la web [38] y que ha sido entregado como material complementario a este documento.

N°	Generadores seleccionados SEN	P_{iny} [MW]	Tipo	Barra de conexión	$P_{iny\ eq.}$ [MW]
1	Conejo Solar	97,50	solar	Paposo	253,07
2	Pampa Solar Norte	69,27	solar	Paposo	
3	Lalackama I y II	67,86	solar	Paposo	
4	Parque Eólico Taltal	18,44	eólica	Paposo	
5	PV Salvador	62,56	solar	Diego de Almagro	62,56
6	Luz del Norte	129,75	solar	Carrera Pinto	212,08
7	Solar Carrera Pinto	82,33	solar	Carrera Pinto	
8	Solar Llano de Llampos	65,26	solar	Cardones	83,07
9	Parque Fotovoltaico Los Loros	17,81	solar	Cardones	
10	Guacolda	222,00	térmica	Maitencillo	222,00
11	El Romero	134,30	solar	Punta Colorada	262,05
12	Eólica San Juan	35,00	eólica	Punta Colorada	
13	El Pelícano	92,75	solar	Punta Colorada	
14	Eólica Los Cururos	28,10	eólica	Las Palmas	48,27
15	El Arrayán	20,17	eólica	Las Palmas	
16	Campiche	272,00	térmica	Nogales	818,00
17	Nueva Ventanas	273,00	térmica	Nogales	
18	Ventanas I y II	273,00	térmica	Nogales	
19	San Isidro I y II	505,00	térmica	Quillota	1445,00
20	Nehuenco I y II	702,00	térmica	Quillota	
21	Quintero	238,00	térmica	Quillota	
22	Quilapilún	79,97	solar	Polpaico	149,57
23	Hornitos	26,20	hidráulica	Polpaico	
24	Los Quilos	23,80	hidráulica	Polpaico	
25	Blanco	19,60	hidráulica	Polpaico	
26	Rapel	117,00	hidráulica	Lo Aguirre	117,00
27	Alfalfal	103,13	hidráulica	Alto Jahuel	190,33
28	Chacayes	45,70	hidráulica	Alto Jahuel	
29	Queltehues	41,50	hidráulica	Alto Jahuel	
30	La Higuera	68,10	hidráulica	Tinguiririca	210,80
31	La Confluencia	60,30	hidráulica	Tinguiririca	
32	Sauzal	51,00	hidráulica	Tinguiririca	
33	San Andrés	31,40	hidráulica	Tinguiririca	

34	Pehuenche	248,00	hidráulica	Ancoa	
35	Colbún	231,00	hidráulica	Ancoa	
36	Viñales	30,90	térmica	Ancoa	556,90
37	Loma Alta	27,00	hidráulica	Ancoa	
38	Machicura	20,00	hidráulica	Ancoa	
39	Cipreses	97,00	hidráulica	Itahue	
40	Curillinque	66,00	hidráulica	Itahue	215,00
41	Isla	52,00	hidráulica	Itahue	
Total P_{iny} [MW]		4845,70			

Tabla 5.2: Selección de plantas con inyecciones más importantes en el estado escogido del sistema según el criterio de "generación horaria" divulgado en el portal de información pública del Coordinador Eléctrico Nacional [38].

Con los datos de producción y consumo ya reclutados, fue posible discriminar las líneas más importantes y, con ello, dibujar un boceto del modelo simplificado de la red. Luego, se recurrió al software DigSilent PF para proyectar el diagrama unilineal del SEN reducido y depositar los datos de generación y demanda bajados desde la plataforma del Coordinador [38], así como los parámetros de las líneas descargados desde el portal "Energía Abierta" de la Comisión Nacional de Energía [44]. En la *Figura 5.1* se presenta el esquema unifilar de la simplificación del SEN en el cual se aplicarán las metodologías seleccionadas. Debe notarse que, al igual que en los ejemplos mostrados en Anexos en el ítem 8.4, se ha conectado un Sistema Equivalente para asumir parte de las inyecciones o retiros que necesite el modelo con el fin de que los datos de generación y demanda ingresados al programa no sufran alteraciones importantes.

Tenga en cuenta que el SEN reducido de la *Figura 5.1* es el esquema que se utilizará para ejecutar los flujos de carga horarios que simularán la obtención de datos desde el servidor del Coordinador; cuyo propósito es la aplicación del Método de Bialek de manera "instantánea". En ese sentido, de acuerdo con los supuestos con los que se hizo la simplificación de la red de transmisión nacional, los flujos de carga horario se efectuarán con la información de estado del sistema del día 28 de febrero de 2018 y, conforme a la curva de producción de PV Salvador, se optó por que asuman el rango de tiempo comprendido entre las 08:00 hrs. y las 21:00 hrs. de dicho día.

En el ítem 8.5.1 de la sección Anexos, se puede hallar la *Tabla A.5.1*, *Tabla A.5.2* y *Tabla A.5.3* que muestran, respectivamente, los datos de entrada, los resultados del flujo de carga y el balance nodal del SEP para las 14:00 hrs. de la fecha acordada. La corrida de flujo fue ejecutada en DigSilent PF y el esquema con el flujo de carga AC del estado analizado también está disponible en la sección Anexos, al final de este trabajo. El archivo ".pfd", en tanto, está adjunto como material complementario a este documento.

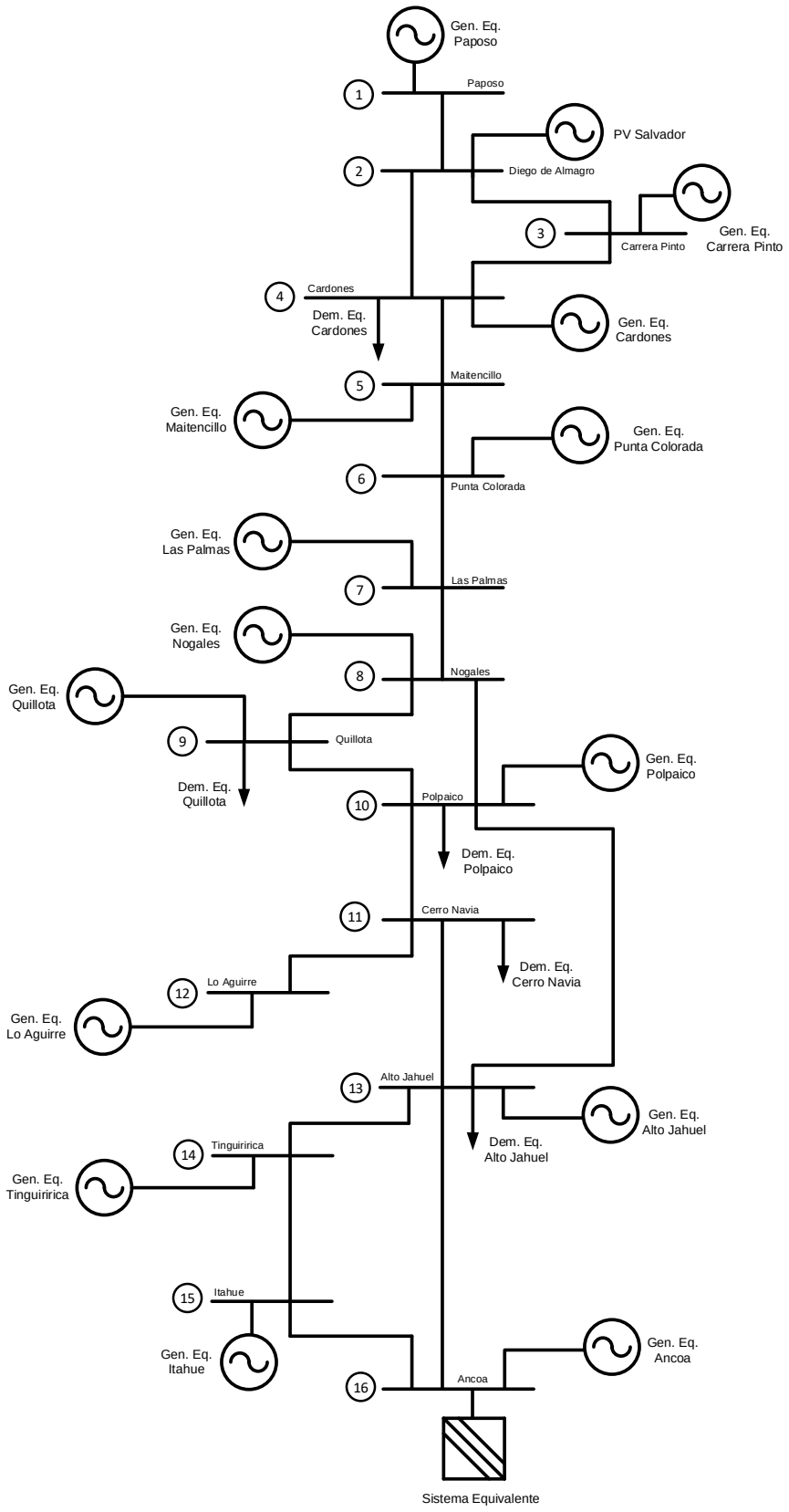


Figura 5.1: Simplificación del Sistema Eléctrico Nacional bajo los criterios del problema de trazabilidad. Elaboración propia.

En lo que sigue, se empleará el Método de Bialek con Flujos Gruesos sobre el SEN simplificado de la *Figura 5.1* con el fin de mostrar la construcción teórica de una *cadena de suministro* a nivel de transmisión nacional. Luego, se aplicará el Método de Prorrateo o Estampillado Postal sobre la barra de Cerro Navia, donde se supondrá un consumo cualquiera en baja tensión para conformar la *cadena de suministro* final. En este punto, el lector debe entender que lo anterior constituirá sólo un ejemplo de los 14 posibles estados del sistema y será la base teórica para comprender la operación de la *aplicación descentralizada*.

5.1.2 Aplicación de Flujos Gruesos de Bialek en modelo simplificado del SEN sujeto al estado de la red del 28 de febrero de 2018 a las 14:00 hrs.

Como se describió en el Capítulo 4, J. Bialek [1] estableció que si lo que se desea es responsabilizar a las cargas por el grado de uso que hacen estas del sistema de transmisión, el camino correcto sería la aplicación del *principio de repartición proporcional* con enfoque en la contribución de potencia inyectada por los generadores para componer el flujo retirado por una carga específica. En coherencia a lo anterior, se debe aplicar el *algoritmo de inyecciones de potencia* revisado en el ítem 4.1.1 sobre el esquema de la *Figura 5.1* y, en consecuencia, conocer las *demandas gruesas* que adoptarán las pérdidas según el estado del sistema.

Para exponer la trazabilidad de flujos a nivel de transmisión nacional, en lo que sigue se mostrará la aplicación del Método de Bialek sobre el modelo simplificado del SEN sujeto a las condiciones de estado de las 14:00 hrs. del 28/02/2018. Al igual que con los ejemplos analizados para el Capítulo 4, el cálculo de pérdidas sobre el circuito se ejecutó con un archivo MATLAB que extrae la información de entrada desde una planilla Excel y que graba los datos de salida sobre la misma (ver código en ítem 8.5.3 en Anexos); existiendo una planilla para cada escenario, esto es, desde las 08:00 hrs. a 21:00 hrs. de la fecha convenida. Los correspondientes archivos “.m” y “.xlsx” de los 14 estados horarios han sido entregados y se pueden consultar en la carpeta titulada “SEN reducido”, alojada en la carpeta “Recursos de software” del material complementario de este trabajo.

5.1.2.1 Resultados

Debido a las dimensiones del cuadro de resultados, en el ítem 8.5.2 de la sección Anexos se presenta la *Tabla A.5.4*, que exhibe la composición de origen de los flujos demandados por las 5 cargas del modelo simplificado del SEN para la hora y fecha convenidas. De todas formas, en función de los objetivos de esta Memoria, los resultados más significativos se exponen a continuación.

Barra	Demanda Gruesa [MW]	Contribución de PV Salvador (G_2) a las Demandas Gruesas del SEN reducido [MW]	Contribución relativa [%]
1	0,00	0,00	0,0%
2	0,00	0,00	0,0%
3	0,00	0,00	0,0%
4	172,23	17,64	28,2%
5	0,00	0,00	0,0%
6	0,00	0,00	0,0%
7	0,00	0,00	0,0%
8	0,00	0,00	0,0%
9	840,38	9,79	15,7%
10	187,02	2,58	4,1%
11	2231,90	23,52	37,6%
12	0,00	0,00	0,0%
13	1397,04	9,03	14,4%
14	0,00	0,00	0,0%
15	0,00	0,00	0,0%
16	17,14	0,00	0,0%
Total [MW]	4845,698	62,56	100%

Tabla 5.3: Aportes de la producción de PV Salvador (G_2) al flujo grueso retirado en Cardones ($\mathcal{D}_4$), Quillota ($\mathcal{D}_9$), Polpaico ($\mathcal{D}_{10}$), Cerro Navia ($\mathcal{D}_{11}$) y Alto Jahuel ($\mathcal{D}_{13}$) a las 14:00 hrs. del 28 de febrero de 2018 en modelo reducido del SEN. Elaboración propia.

Asignación parcial de pérdidas a consumidores del SEN reducido			
Barra	**Demanda Gruesa [MW]**	**Demanda Real [MW]**	**Pérdidas por barra [MW]**
1	0,00	0,00	0,00
2	0,00	0,00	0,00
3	0,00	0,00	0,00
4	172,23	168,39	3,84
5	0,00	0,00	0,00
6	0,00	0,00	0,00

7	0,00	0,00	0,00
8	0,00	0,00	0,00
9	840,38	812,60	27,78
10	187,02	179,25	7,77
11	2231,90	1979,16	252,74
12	0,00	0,00	0,00
13	1397,04	1368,05	28,99
14	0,00	0,00	0,00
15	0,00	0,00	0,00
16	17,14	17,13	0,01
Total	4845,70	4524,58	321,12

Tabla 5.4: Repartición de pérdidas entre las cargas del SEN reducido conforme al Método de Bialek empleando Flujos Gruesos. La asignación por barra corresponde a la diferencia entre la *demanda gruesa* y *real* en este estado del sistema. Elaboración propia.

Contribuciones del parque generador del SEN reducido a la demanda en Cerro Navia ($\mathcal{D}_{11}$)		
Generador	**Abastecimiento barra 11 [MW]**	**Contribución relativa [%]**
G1	95,13	4,26%
G2	23,52	1,05%
G3	79,72	3,57%
G4	31,23	1,40%
G5	116,22	5,21%
G6	137,19	6,15%
G7	25,27	1,13%
G8	428,23	19,19%
G9	665,86	29,83%
G10	100,13	4,49%
G11	-	0,00%
G12	117,00	5,24%
G13	67,91	3,04%
G14	74,73	3,35%
G15	75,14	3,37%
G16	194,63	8,72%
Total	2231,90	100%

Tabla 5.5: Composición de origen del flujo retirado en la barra Cerro Navia ($\mathcal{D}_{11}$) del modelo simplificado del SEN (*Figura 5.1*) el 28/02/2018 a las 14:00 hrs. según el Método de Bialek y en coherencia a los supuestos planteados. Elaboración propia.

5.1.2.2 Análisis general

De las tablas que anteceden, y en consistencia a lo comentado en el Capítulo 4, los resultados de la aplicación del Método de Bialek con Flujos Gruesos permiten concluir que se atribuye mayor responsabilidad a aquellas cargas que hacen un mayor uso del sistema de transmisión en virtud de la composición de origen de la potencia consumida, esto es, según la procedencia y distribución de los flujos de potencia por la red de transporte. Esta conclusión es independiente de las cantidades retiradas y establece una dependencia entre la asignación de pérdidas y la ubicación del *pool* de productores que efectivamente abastecen la demanda de una barra específica.

Así, si una carga no puede alimentarse completamente desde un generador conectado en su misma barra, esta deberá recurrir a generadores ubicados en otros puntos de la red para cubrir su demanda, incrementando el costo físico que tendrá que asumir por ello. En consecuencia, cuantas más pérdidas produzcan las líneas que transportan los flujos que la abastecen, tanto más aumentará su responsabilidad por concepto de uso del sistema de transmisión. Hay que enfatizar que en esta última aseveración tienen especial relevancia los conceptos de cargabilidad (donde influyen de manera importante tanto los parámetros eléctricos del cableado como las tensiones de operación) y de congestión del transporte de energía (donde la cantidad de agentes que hacen uso de una misma línea puede ser un aspecto crítico).

Si bien los agentes del sistema de trazabilidad que se va a desarrollar en la *aplicación descentralizada* están conectados en las barras Diego de Almagro y Cerro Navia, el Método de Bialek involucra la participación del resto de los generadores y demandas de la red, por lo cual también es bueno revisar someramente cuáles son los productores y consumidores más relevantes en este estado del sistema.

En ese sentido, la *Tabla 5.3* arroja la presencia de abastecimiento que tiene PV Salvador (G_2) sobre las cargas del SEN reducido; donde se observa que la planta destina la mayoría de su producción a alimentar las *demandas gruesas* de Cerro Navia (barra 11, $\mathcal{D}_{11}$), Cardones (barra 4, $\mathcal{D}_4$) y Quillota (barra 9, $\mathcal{D}_9$), en orden decreciente. No obstante, se debe contemplar que existen otros generadores cuyo servicio queda limitado al consumo local, y que, por razones de producción o de limitación de las líneas, no pueden hacer aportes sobre el total de cargas del sistema (como demuestra G_{12} conectado en Lo Aguirre, que entrega el total de su generación a Cerro Navia, por ejemplo). Lo anterior es una muestra de que, bajo el Método de Flujos Gruesos de Bialek, la ubicación de las barras de inyección constituye un factor importante.

Otro punto interesante es la diversificación de fuentes de suministro que tienen algunas cargas. En efecto, si se pone atención a la *demanda gruesa* de Cardones y Polpaico (barra 10, $\mathcal{D}_{10}$) mostradas en la *Tabla*

A.5.4 del ítem 8.5.2 de la sección Anexos, se percibirá que el consumo es similar en magnitud ($172{,}23[MW]$ y $187{,}02[MW]$, correspondientemente) y que ambas cargas se alimentan con flujos que viajan de norte a sur, pero que, a diferencia de $\mathcal{D}_4$, la demanda $\mathcal{D}_{10}$ se abastece de un número más amplio de generadores, a saber: $\{G_1, \dots G_j \dots, G_{10}\}$; $j \in \mathbb{N}$.

La *Tabla A.5.4* también muestra que la contribución relativa de PV Salvador es parecida a la que tiene el generador equivalente G_7 en Las Palmas. Esto justifica que los aportes a la *demanda gruesa* en Cerro Navia no dependen exclusivamente de la longitud entre los puntos de generación y consumo, ya que el énfasis estaría puesto en la ubicación de la producción energética (punto de inyección o procedencia) y en la capacidad de transferencia de los flujos hacia los puntos de retiro (distribución a través de la red).

En otro tema, la *Tabla 5.4* muestra las pérdidas que debe adoptar cada carga del SEN reducido, dejando ver que la demanda en Cerro Navia es la que efectúa el retiro más importante y, en este caso, es quien se adjudica el costo físico mayor. La segunda asignación más importante es para la carga conectada en Alto Jahuel (barra 13, $\mathcal{D}_{13}$) y en tercer lugar está la demanda en Quillota, donde las pérdidas atribuidas llegan a los $27{,}78[MW]$. Aquí resulta interesante notar que $\mathcal{D}_{13}$ asume una magnitud parecida de pérdidas a las tomadas por $\mathcal{D}_9$, aún cuando en Alto Jahuel se efectúa casi un 67% más de retiros que en Quillota. Esto ilustra que la distribución de las pérdidas en el sistema no se basa en las cantidades demandadas, sino que en el camino que recorren los flujos para alimentar dichas cargas; por lo que se puede inferir que el uso de las líneas que une cada consumo con su respectiva generación, en este caso, es similar. Lo anterior es poco evidente, y más aún, si se compara la cuantía y proveniencia de los flujos que alimentan a $\mathcal{D}_9$ con el respectivo conjunto que abastece a $\mathcal{D}_{13}$ (ver *Tabla A.5.4* del ítem 8.5.2, en sección Anexos).

También, los datos agrupados en la *Tabla A.5.4* revelan que los aportes más sobresalientes sobre las grandes demandas del SEN reducido (digamos, $\mathcal{D}_{11}$ y $\mathcal{D}_{13}$) provienen de generadores con altos niveles de producción que no necesariamente están conectados de forma directa con la barra alimentada. Al respecto, y particularmente para Cerro Navia, la *Tabla 5.5* muestra que para cubrir el retiro efectuado por $\mathcal{D}_{11}$ existe la participación de todos los generadores del sistema; siendo G_9 en Quillota quien realiza la mayor contribución en el estado analizado, llegando a los $665{,}86[MW]$ y abasteciendo el 29,8% de la demanda local. Los aportes que siguen son el de G_8 en Nogales (barra 8) y el de G_{16} en Ancoa (barra 16), y ninguno de ellos está conectado de forma inmediata a la barra 11.

Por último, es bueno recordar que la participación de cada generador en el abastecimiento y las pérdidas de cada *demanda gruesa*, corresponden a cantidades que están sujetas al estado del SEP sobre el que se trabaja y, por lo tanto, pueden presentar importantes variaciones en otro escenario de operación del sistema.

5.1.2.3 Construcción de cadena de suministro a nivel nacional

De acuerdo con la aplicación de los Flujos Gruesos de Bialek en el modelo simplificado del SEN (ver *Figura 5.1*), y en función de la *Tabla 5.3* y *Tabla 5.5*, conviene tener en cuenta los siguientes resultados:

Asignación de pérdidas a barra Cerro Navia	
Demanda Gruesa ($\mathcal{D}_{11}$) [MW]	2231,90
Demanda Real ($\mathcal{D}_{11}^{real}$) [MW]	1979,16
Pérdidas en barra 11 [MW]	252,74

Tabla 5.6: Pérdidas locales en Cerro Navia en modelo simplificado del SEN para el 28/02/2018 a las 14:00 hrs. Corresponde al costo físico de transportar y distribuir los flujos generados desde las plantas hacia la barra 11. Elaboración propia.

Contribución de Generación Gruesa a las pérdidas en Cerro Navia		
Generador	**Contribución relativa [%]**	**Pérdidas [MW]**
G1	4,26%	10,77
G2	1,05%	2,66
G3	3,57%	9,03
G4	1,40%	3,54
G5	5,21%	13,16
G6	6,15%	15,53
G7	1,13%	2,86
G8	19,19%	48,49
G9	29,83%	75,40
G10	4,49%	11,34
G11	0,00%	-
G12	5,24%	13,25
G13	3,04%	7,69
G14	3,35%	8,46
G15	3,37%	8,51
G16	8,72%	22,04
Total	100,0%	252,74

Tabla 5.7: Composición de origen de las pérdidas locales asignadas a Cerro Navia ($\mathcal{D}_{11}$) según Flujos Gruesos de Bialek. Aplicación sobre modelo simplificado del SEN para los supuestos y el estado estudiados (*Figura 5.1*). Elaboración propia.

La *Tabla 5.7* enseña que, tal como sugiere J. Bialek en su "paper" [1], la estructura de las pérdidas asignadas a una carga guarda las mismas proporciones que las de abastecimiento (ver *Tabla 5.5*). Esto quiere decir que los niveles de participación del *pool* de generadores que alimentan a una *demanda gruesa* se mantienen

constantes para la *demanda real* y para las pérdidas. En efecto, de la *Tabla 5.5* se desprende que la planta PV Salvador aporta con $23{,}52[MW]$ en Cerro Navia, siendo equivalente al 1,05% del flujo retirado en dicha barra y representando el 37,6% de la producción total de G_2 (ver *Tabla 5.3*). A su vez, el costo físico que debe asumir la *demanda gruesa* $\mathcal{D}_{11}$ por concepto de transporte de flujos provenientes desde G_2, es $2{,}66[MW]$ (ver *Tabla 5.7*); siendo equivalente, también, al 1,05% de las pérdidas locales en Cerro Navia.

Con lo anterior, si se denota por "G_{2-11}" a la porción de potencia que destina PV Salvador a Cerro Navia, por "$\mathcal{D}_{11-2}^{real}$" al abastecimiento *real* máximo que puede consumir Cerro Navia desde PV Salvador y por "$P_{losses_{11-2}}$" a la fracción de pérdidas locales correspondiente a esta transferencia energética (ver *Tabla 5.7*), se consigue el siguiente balance:

$$\Rightarrow G_{2-11} = P_{losses_{11-2}} + \mathcal{D}_{11-2}^{real}$$
$$\Leftrightarrow 23{,}52[MW] = 2{,}66[MW] + \mathcal{D}_{11-2}^{real}$$
$$\therefore\ \mathcal{D}_{11-2}^{real} = 20{,}86[MW]$$

Advierta que esta cifra representa el total a repartir entre todos los clientes finales que se alimentarán exclusivamente de las inyecciones de PV Salvador (algo así como los clientes que tendrán un contrato de suministro con dicha planta), lo que a nivel nacional puede representarse de la siguiente forma:

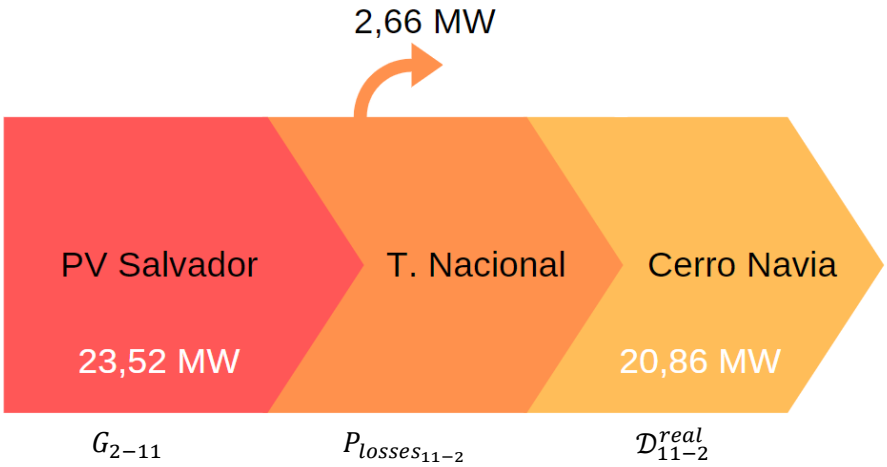


Figura 5.2: Cadena de suministro a nivel de transmisión nacional empleando Flujos Gruesos de Bialek sobre el SEN reducido de la *Figura 5.1*. Cálculo efectuado sobre la barra Cerro Navia para el 28/02/2018 a las 14:00 hrs. Elaboración propia.

La infografía anterior refleja la trazabilidad de flujos desde la planta PV Salvador hasta la barra Cerro Navia estableciendo la producción, pérdidas y demanda admisibles para el estado y modelo planteado. En otras palabras, la *cadena de suministro* a nivel de transmisión nacional refleja el balance de flujos instantáneo

arrojado por el Método de Bialek para validar las cuantías de origen, de uso de transporte y de la capacidad de abastecimiento en la barra alimentadora de Santiago. Si se deseara conocer las líneas del SEN reducido sobre las que circula el flujo solicitado por $\mathcal{D}_{11}$, la metodología lo puede despejar, pero dicho concepto de rastreo de flujos escapa a los objetivos de este trabajo y, en su lugar, se ha optado por mostrar el costo físico de transportar las inyecciones hacia la barra que abastece al cliente final.

5.1.3 Aplicación de Estampillado Postal en modelo simplificado del SEN sujeto al estado de la red del 28 de febrero de 2018 a las 14:00 hrs.

Como se explicó en el Capítulo 4, con el Método de Prorrateo o Estampillado Postal es posible conocer el costo físico de transporte atribuible a una carga puntual conforme a la magnitud del retiro efectuado, independientemente del lugar de la red en el que se producen las pérdidas. En ese sentido, si se hace caso omiso de la ubicación de los puntos de producción, de consumo y de las líneas que "viven dentro" de la carga $\mathcal{D}_{11}$ del SEN reducido de la *Figura 5.1* (esto es, a nivel de transmisión zonal y distribución), y en su lugar se trabaja con el supuesto de que cada consumo se abastece de flujos que utilizan la totalidad de las instalaciones [45], entonces se puede entender que el cliente final está inmerso en "una gran bolsa de energía".

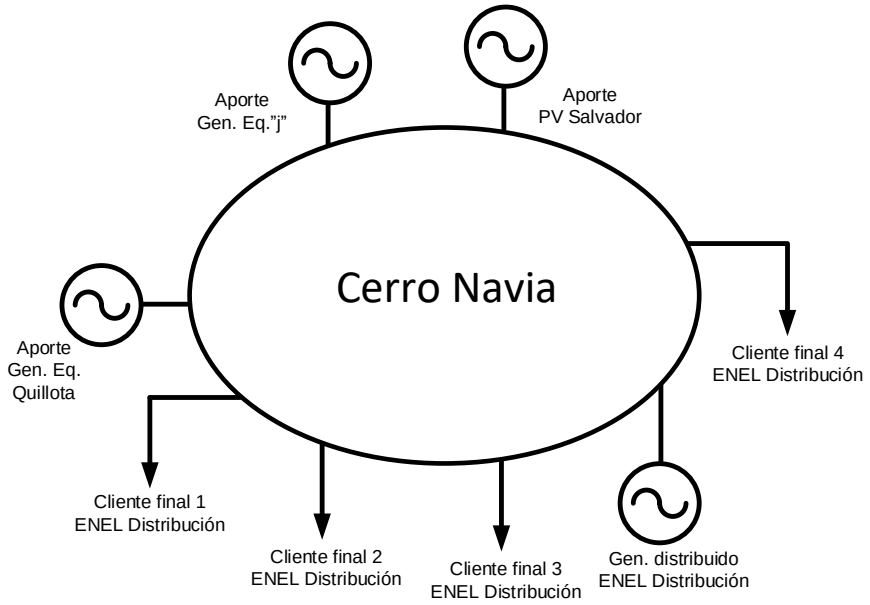


Figura 5.3: Esquema del sistema de transmisión zonal y de distribución bajo la aplicación del Método de Prorrateo o Estampillado Postal en la carga $\mathcal{D}_{11}$ del modelo eléctrico de la *Figura 5.1*. Elaboración propia.

La imagen anterior representa el sistema cerrado en el que coexisten todos los agentes que se alimentan desde la barra Cerro Navia y también quienes la abastecen, que en función de la *Tabla 5.5*, es sabido que

se compone de la participación de todo el parque generador. Sin embargo, a modo de ejemplo, se ha simbolizado la contribución de algunas plantas además de la presencia de cuatro clientes finales. Observe que explicitar las conexiones de líneas y barras no tiene cabida, puesto que cada carga del sistema utiliza, en distinta proporción, el total de instalaciones dispuestas para el transporte de flujos en media y baja tensión. Con ello, es irrelevante si el consumidor final está conectado en transmisión zonal o en distribución.

5.1.3.1 Resultados

Dado que este procedimiento de asignación de pérdidas no necesita flujos de carga, pero en este caso se combinará de forma sinérgica con el Método de Bialek, se usará el esquema que antecede para ejemplificar la aplicación del método en perspectiva de conformar la *cadena de suministro* de un cliente final cuando el SEN reducido opera a las 14:00 hrs. del 28 de febrero de 2018. Al respecto, dado que se está en conocimiento del retiro máximo que Cerro Navia puede efectuar desde PV Salvador, así como de las pérdidas locales atribuibles a dicha barra (vía Flujos Gruesos), la *Tabla 5.8* muestra la parte que falta: la asignación de pérdidas y los retiros efectuados por los cuatro clientes ficticios de la *Figura 5.3*.

Asignación de pérdidas a clientes finales en Cerro Navia			
Agente	**Retiro [MW]**	**Proporción**	**Prorrateo [MW]**
Cliente final 1	2,086	10%	0,266
Cliente final 2	8,344	40%	1,064
Cliente final 3	6,258	30%	0,798
Cliente final 4	4,172	20%	0,532
Total	20,86	100%	2,66

Tabla 5.8: Repartición proporcional de las pérdidas locales en Cerro Navia según criterios de la *Figura 5.3*. Elaboración propia.

5.1.3.2 Análisis General

La *Tabla 5.8* expone cómo se raciona la contribución de PV Salvador en Cerro Navia. Recuerde que para llegar hasta aquí primero fue necesario reconocer la participación de G_2 en el abastecimiento a la *demanda gruesa* $\mathcal{D}_{11}$ (de 1,05% según la *Tabla 5.5*). Luego, fue posible despejar las pérdidas atribuibles en transmisión nacional y, con ello, se pudo conocer la parte del flujo total retirado en Cerro Navia abastecido exclusivamente desde Diego de Almagro ($\mathcal{D}_{11-2}^{real} = 20{,}86[MW]$, *demanda real*, sin pérdidas). Finalmente, en este ítem ha sido posible prorratear la "inyección solar" y las "pérdidas solares" procedentes desde PV Salvador entre los cuatro clientes finales de Santiago, conservando las mismas proporciones de repartición según el consumo de estos. Por último, tal como se comentó en el ítem 4.3 del capítulo anterior y como lo

corrobora la *Tabla 5.8*, se debe decir que bajo el Método de Estampillado Postal los agentes que retiran más potencia son, también, los que producen mayor cantidad de pérdidas eléctricas en la red.

5.1.3.3 Construcción de cadena de suministro a nivel zonal y de distribución

Hasta aquí debe estar claro qué parte de la generación de G_2 se destina a $\mathcal{D}_{11}$ (ver *Tabla 5.3*), cuál es el nivel de participación de G_2 en la cobertura de $\mathcal{D}_{11}$ (ver *Tabla 5.5*), cuántas son las pérdidas a nivel de transmisión nacional por este concepto (ver *Tabla 5.7*) y cuál es la *demanda real* alimentada por G_2 que se repartirá entre los clientes finales ($\mathcal{D}_{11-2}^{real}$). Así, si se denota por "k_{G_2}" a la producción relativa que PV Salvador destina a Cerro Navia (37,6% en *Tabla 5.3*) y si se denota por "$k_{\mathcal{D}_{11}}$" al factor de proporción con el que se prorratean las pérdidas locales en la misma barra (1,05% en *Tabla 5.7*), se tiene la siguiente igualdad teórica a nivel de transmisión nacional:

$$\Rightarrow G_{2-11} = P_{losses_{11-2}} + \mathcal{D}_{11-2}^{real}$$

$$\Leftrightarrow k_{G_2} \cdot G_2 = k_{\mathcal{D}_{11}} \cdot P_{losses_{11}} + \mathcal{D}_{11-2}^{real}$$

$$\boxed{\therefore\ k_{G_2} \cdot G_2 = \left(\frac{k_{G_2} \cdot G_2}{\mathcal{D}_{11}}\right) \cdot P_{losses_{11}} + \mathcal{D}_{11-2}^{real}}$$

Ecuación 5.1: Expresión de *cadena de suministro* usando Flujos Gruesos de Bialek a nivel nacional. Elaboración propia.

Luego, si se denota por "$k_{\mathcal{D}_{11-2}}$" al factor de proporción con el que se prorratean los consumos y las pérdidas a nivel de transmisión zonal y distribución dentro de Cerro Navia, se consigue:

$$\Rightarrow k_{\mathcal{D}_{11-2}} \cdot k_{G_2} \cdot G_2 = k_{\mathcal{D}_{11-2}} \cdot \left(\left(\frac{k_{G_2} \cdot G_2}{\mathcal{D}_{11}}\right) \cdot P_{losses_{11}} + \mathcal{D}_{11-2}^{real}\right)$$

$$\Leftrightarrow \frac{\mathcal{D}_{Cliente}}{\mathcal{D}_{11-2}^{real}} \cdot k_{G_2} \cdot G_2 = \frac{\mathcal{D}_{Cliente}}{\mathcal{D}_{11-2}^{real}} \cdot \left(\left(\frac{k_{G_2} \cdot G_2}{\mathcal{D}_{11}}\right) \cdot P_{losses_{11}} + \mathcal{D}_{11-2}^{real}\right)$$

Por lo que, la *cadena de suministro* instantánea a nivel del cliente final vendrá definida por la relación:

$$\boxed{\therefore\ \frac{\mathcal{D}_{Cliente}}{\mathcal{D}_{11-2}^{real}} \cdot k_{G_2} \cdot G_2 = \frac{\mathcal{D}_{Cliente}}{\mathcal{D}_{11-2}^{real}} \cdot \left(\frac{k_{G_2} \cdot G_2}{\mathcal{D}_{11}}\right) \cdot P_{losses_{11}} + \mathcal{D}_{Cliente}}$$

Ecuación 5.2: Expresión de *cadena de suministro* a nivel de cliente *real* usando Flujos Gruesos y Prorrateo. Elaboración propia.

La expresión anterior dice que cuando un consumidor conectado a Cerro Navia efectúa un retiro desde la red en una proporción "*x*" respecto del total local, este se responsabiliza de la porción "*x*" de las pérdidas locales adjudicadas, así como de la fracción "*x*" de suministro procedente desde PV Salvador.

Como se verá a partir del ítem 5.2, la *Ecuación 5.2* es relevante, ya que será incorporada en la *aplicación descentralizada* y, para conseguir dicha igualdad en términos de implementación, dado que G_2 y $\mathcal{D}_{Cliente}$ serán variables empíricas medidas por los nodos de *Blockchain GTIME*, gracias a la *Ecuación 5.1* bastará con conocer el factor de contribución k_{G_2} de PV Salvador, la *demanda gruesa* $\mathcal{D}_{11}$ en Cerro Navia y la *demanda real* $\mathcal{D}_{11}^{real}$ en la misma barra (ver *Tabla 5.6*).

A modo de ejemplo, si se usa el consumo del "Cliente final 1" de la *Tabla 5.8* ($\mathcal{D}_{Cliente1} = 2{,}086[MW]$) para enseñar la construcción teórica de la *cadena de suministro* instantánea de electricidad entre PV Salvador y un cliente final en Santiago, se obtiene:

<u>Trazabilidad de flujos a nivel de transmisión nacional (ver *Tabla 5.3*, *Tabla 5.6* y *Tabla 5.7*):</u>

$$\Rightarrow k_{G_2} \cdot G_2 = k_{\mathcal{D}_{11}} \cdot P_{losses_{11}} + \mathcal{D}_{11-2}^{real}$$

$$\Leftrightarrow k_{G_2} \cdot G_2 \ = \left(\frac{k_{G_2} \cdot G_2}{\mathcal{D}_{11}}\right) \cdot P_{losses_{11}} + \mathcal{D}_{11-2}^{real}$$

$$\Leftrightarrow 0{,}376 \cdot 62{,}56 \ = \left(\frac{0{,}376 \cdot 62{,}56}{2231{,}9}\right) \cdot (2231{,}9 - 1979{,}16) + \mathcal{D}_{11-2}^{real}$$

$$\Leftrightarrow 23{,}52 \ = \ 2{,}66 \ + \ \mathcal{D}_{11-2}^{real} \quad \Rightarrow \mathcal{D}_{11-2}^{real} = 20{,}86[MW]$$

$$\boxed{\therefore G_{2-11} = 23{,}52[MW] \quad ; \quad P_{losses_{11-2}} = 2{,}66[MW] \quad ; \quad \mathcal{D}_{11-2}^{real} = 20{,}86[MW]}$$

<u>Trazabilidad de flujos a nivel de transmisión zonal y de distribución (ver *Tabla 5.8*):</u>

$$\Rightarrow \frac{\mathcal{D}_{Cliente1}}{\mathcal{D}_{11-2}^{real}} \cdot k_{G_2} \cdot G_2 \ = \frac{\mathcal{D}_{Cliente1}}{\mathcal{D}_{11-2}^{real}} \cdot \left(\frac{k_{G_2} \cdot G_2}{\mathcal{D}_{11}}\right) \cdot P_{losses_{11}} + \mathcal{D}_{Cliente1}$$

$$\Leftrightarrow \frac{2{,}086}{20{,}86} \cdot 23{,}52 \ = \frac{2{,}086}{20{,}86} \cdot 2{,}66 + 2{,}086$$

$$\Leftrightarrow 2{,}352 \ = 0{,}266 + 2{,}086$$

$$\boxed{\therefore G_{Cliente1} = 2{,}352[MW] \quad ; \quad P_{losses_{Cliente1}} = 0{,}266[MW] \quad ; \quad \mathcal{D}_{Cliente1} = 2{,}086[MW]}$$

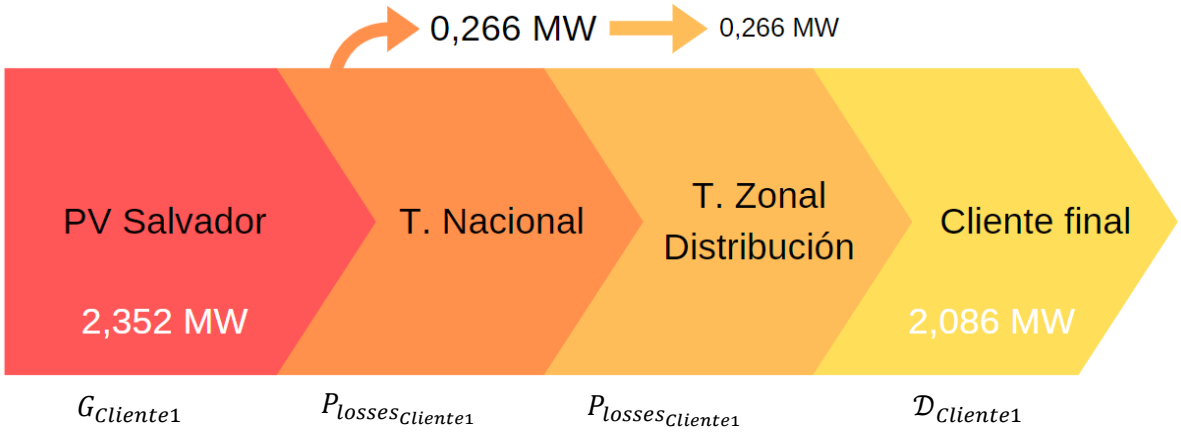


Figura 5.4: Cadena de suministro para una carga *real* alimentada desde la barra Cerro Navia del SEN reducido de la *Figura 5.1*. Resultado de la aplicación de Flujos Gruesos a nivel nacional y de Estampillado Postal a nivel zonal y de distribución para el "Cliente final 1" (ver *Figura 5.3* y *Tabla 5.8*). Se observa la producción destinada al cliente y las pérdidas proporcionales según su nivel de consumo. Elaboración propia.

La imagen que antecede representa la *cadena de suministro* a nivel del cliente final para una carga que se alimenta de las inyecciones que llegan a la barra Cerro Navia, siendo el formato de infografía que se pretende implementar en la *aplicación descentralizada*. El esquema exhibe la trazabilidad del flujo que nace con la producción energética, que viaja por el bloque de transporte desplazando el costo físico hacia la demanda y que culmina en el retiro final efectuado por el cliente. Advierta que, en sintonía a la *Ecuación 5.2*, aun cuando las pérdidas locales a nivel de transmisión nacional ya han sido adjudicadas mediante Flujos Gruesos, la *cadena de suministro* a nivel del cliente final ha tomado las cuantías de todas las etapas del abastecimiento y las ha repartido de forma proporcional al retiro, o equivalentemente, bajo los preceptos del Estampillado Postal.

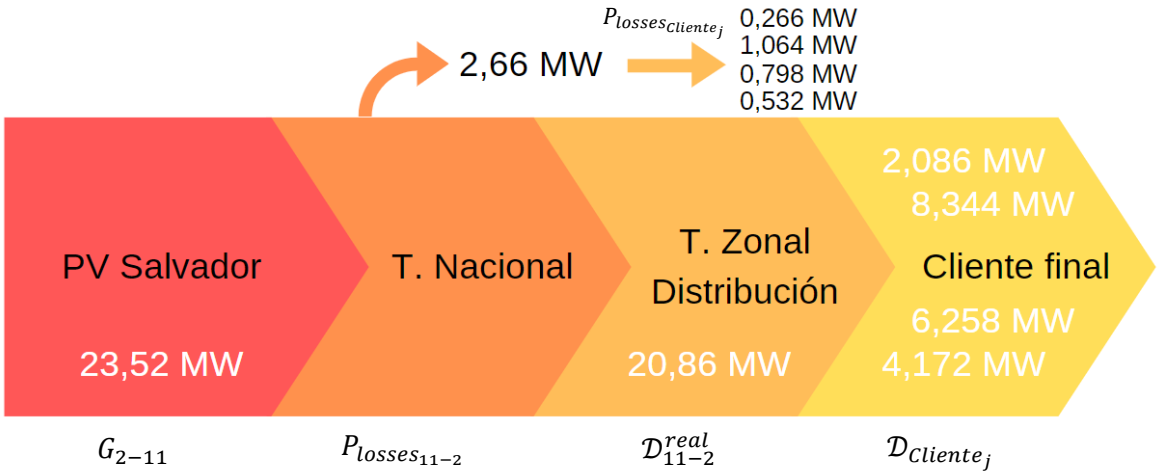


Figura 5.5: Cadena de suministro a nivel de transmisión nacional, zonal y de distribución en carga $\mathcal{D}_{11}$ del modelo simplificado del SEN (*Figura 5.1*). Se distingue la cuantía de producción que PV Salvador destina a Cerro Navia, así como el resultado de prorratear la *demanda real* y las pérdidas locales según el nivel de consumo de los clientes finales (*Tabla 5.8*). Elaboración propia.

Para reforzar la idea anterior se muestra la *Figura 5.5*, que exhibe la composición global de la *cadena de suministro* en todos los niveles de transporte (transmisión nacional, zonal y distribución) a las 14:00 hrs. La ilustración tiene como finalidad resumir las cuantías más relevantes de la interacción energética entre la planta fotovoltaica y el consumidor final a las 14:00 hrs., evidenciando el total de la producción destinada a alimentar Cerro Navia y el "pago" físico a nivel nacional que debe asumir esta por la solicitud de flujos desde Diego de Almagro. También, se expone el traspaso de pérdidas al bloque de consumo, donde se explicita el desglose proporcional del cargo local entre los cuatro clientes finales. Así, para cada uno de ellos existe una porción de la generación de G_{2-11}, una fracción de las pérdidas eléctricas y un pedazo de la "demanda solar" disponible ($\mathcal{D}_{11-2}^{real}$). Todo, según la combinación virtuosa de Flujos Gruesos y Prorrateo.

5.1.4 Resumen previo a la implementación

Las relaciones matemáticas mostradas previamente utilizan el Método de Bialek para asignar las pérdidas a la barra Cerro Navia y, sobre dicho valor local, se aplica el Método de Estampillado Postal para lograr la trazabilidad de flujos desde la producción al consumo. Para conseguir dicho mecanismo de asignación de pérdidas "mixto", al comenzar la sección 5.1 se hizo hincapié en que fue necesario crear un modelo simplificado del SEN sobre el cual se definieron 14 estados horarios que, dotados de información empírica publicada por el Coordinador, permitieron obtener las pérdidas arrojadas vía Flujos Gruesos. Ante esto el lector debe comprender que contar con la simulación de 14 *cadenas de suministro* a nivel de transmisión nacional permite disponer, no solo de una metodología basada en el *principio de repartición proporcional*, sino de resultados que podrán consultarse de manera instantánea para formular la *cadena de suministro* empírica y "a tiempo real" perseguida en este trabajo. Con lo anterior se quiere dejar claro que, debido a que no se cuenta con un permiso para acceder a los equipos de telemetría del Coordinador ni a un modelo simplificado oficial en tiempo real, para el prototipo presentado en este trabajo se ocuparán los resultados del Método de Bialek calculado en los 14 estados horarios del 28 de febrero de 2018 para emular la obtención de información "instantánea" del SEN. Teniendo esto en cuenta será más fácil entender la gestión de datos de la *aplicación descentralizada*, del *smart contract* y, en definitiva, del sistema de trazabilidad con *blockchain*.

5.2 Desarrollo de Aplicación Descentralizada y programación de Smart Contract

Este segundo ítem tiene como propósito mostrar los aspectos generales de la *aplicación descentralizada* (*DApp*) desarrollada en conjunto con la empresa *Phineal* para un sistema de trazabilidad de energía eléctrica entre la planta PV Salvador y el cliente final en Santiago. En términos generales, la *DApp* se vale de un *smart contract* que gestiona los datos de energía para construir una *cadena de suministro* instantánea de electricidad con el fin de certificar, vía *blockchain*, que el retiro del cliente final ha sido generado total o parcialmente utilizando el recurso solar como fuente energética primaria.

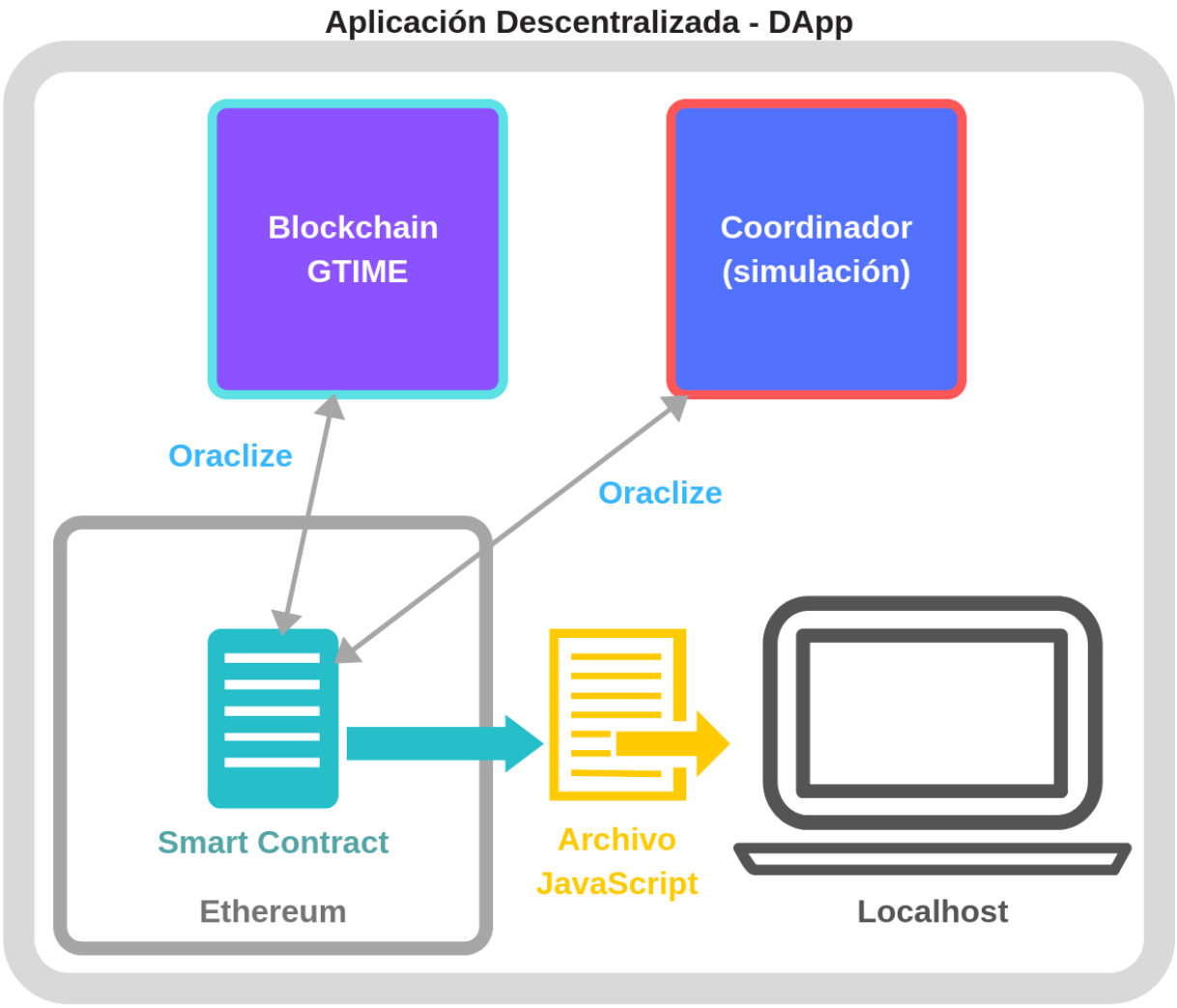


Figura 5.6: Infografía del funcionamiento de la *DApp*. Mediante el despliegue de un *smart contract* en la *blockchain pública* de *Ethereum*, es posible importar las mediciones de inyección y retiro desde la plataforma de certificación solar *Sello Sol* (basada en *Blockchain GTIME*), así como los resultados del Método de Bialek simulados con los datos empíricos del Coordinador. El archivo *JavaScript* recoge la información reclutada en el *smart contract* para emplear el Método de Estampillado Postal y mostrar la *cadena de suministro* instantánea en la web. Elaboración propia.

5.2.1 Descripción general de la DApp

La *Figura 5.6* representa esquemáticamente la interacción de todos los elementos que componen la *DApp*. Lo primero que se debe entender es que el *smart contract* es el gestor de los datos de energía. En términos prácticos, el *smart contract* es un archivo contenido en la carpeta de la *DApp* titulado "`EnergyTraceability.sol`" que está escrito en lenguaje Solidity; que es un lenguaje orientado a objetos diseñado especialmente para la codificación de *contratos inteligentes* y que nace de otros lenguajes tradicionales como C, Python y Java. Como se dijo en el ítem 2.2 de este trabajo, el *smart contract* se debe desplegar en una *blockchain* (red *pública* de *Ethereum*, en este caso) desde donde operará como un nodo más de esta. Luego, las instrucciones lógicas grabadas en el *contrato* se pueden invocar desde cualquier otro nodo de la *blockchain* y, por cada interacción con el *smart contract*, se deberá validar la transacción de *activos digitales* implicados (datos de energía, en este proyecto). En consecuencia, la autoejecución del archivo tiene el propósito de conseguir y derivar, en tiempo real, toda la información para construir las etapas de la transferencia energética.

En segundo lugar, como ilustra la *Figura 5.6*, está la interacción del *smart contract* con dos servidores en los que se encuentran almacenadas las mediciones de energía y las variables de transmisión nacional, respectivamente. En relación a estas últimas, según lo abordado en el ítem 5.1, han sido calculadas previamente mediante la aplicación de Flujos Gruesos de Bialek en 14 estados horarios construidos con información empírica publicada en la web [38]. Luego, en función de la *Ecuación 5.1* y *Ecuación 5.2*, y dado que G_2 y $\mathcal{D}_{Cliente}$ son mediciones proporcionadas por el *Server* de *Blockchain GTIME*, basta con disponer de las variables horarias k_{G_2}, $\mathcal{D}_{11}$ y $\mathcal{D}_{11}^{real}$ dentro del servidor ficticio del Coordinador para que el *smart contract* pueda gestionar los datos que conformarán la *cadena de suministro* instantánea a nivel del cliente final (ver *Tabla 5.9*).

Datos disponibles en servidor ficticio del Coordinador			
Hora	**Factor de contribución k_{G_2} [-]**	**Demanda gruesa Cerro Navia $\mathcal{D}_{11}$ [MW]**	**Demanda real Cerro Navia $\mathcal{D}_{11}^{real}$ [MW]**
8:00	0,0000000	1533,104	1410,565
9:00	0,2297696	1731,400	1569,347
10:00	0,3606879	1928,262	1719,709
11:00	0,3748813	2038,806	1812,082
12:00	0,3748689	2133,475	1895,276
13:00	0,3794214	2199,120	1949,613
14:00	0,3758898	2231,895	1979,160
15:00	0,3761839	2308,552	2017,568
16:00	0,3898875	2368,921	2046,353

17:00	0,3859495	2323,342	2015,006
18:00	0,3842972	2222,447	1945,943
19:00	0,3496875	2043,973	1816,268
20:00	0,0164397	1894,945	1711,653
21:00	0,0000000	1959,267	1758,669

Tabla 5.9: Variables resultantes de la aplicación de Flujos Gruesos en SEN reducido de la *Figura 5.1* con el fin de simular la obtención de una *cadena de suministro* instantánea a nivel de transmisión nacional. Rango horario definido según período de producción de planta PV Salvador para el 28/02/2018. Para el resto de las horas todos los valores son cero. Elaboración propia.

Respecto del cuadro anterior es bueno comentar que, debido a que los datos de inyección y retiro se miden cada 20 minutos (según se explicó en el Capítulo 3), y dado que la frecuencia de muestreo de datos publicada por el Coordinador es horaria [38], las variables simuladas presentadas en la *Tabla 5.9* se utilizarán para construir tres *cadenas de suministro* por hora, naturalmente.

Volviendo al esquema de la *Figura 5.6*, se debe notar que la correspondencia biunívoca del *smart contract* con los servidores de datos energéticos está representada con una flecha de dos sentidos y, en términos prácticos, equivale a la solicitud y recepción de las cuantías de producción y consumo levantadas por los nodos medidores de *Blockchain GTIME* (*phiNet* y *phiNergy*), así como a la solicitud y recepción de información útil para prorratear las pérdidas locales en Cerro Navia (ver *Tabla 5.9*). Dado que cualquier tipo de movimiento de información sobre el *smart contract* implica una transacción descentralizada en *Ethereum* que se debe validar, fue necesario incorporar una herramienta especial para admitir las solicitudes y recepciones de datos; un servicio de "oráculo" que consiguiera importar los datos certificados desde la *Blockchain GTIME* y los datos simulados desde el servidor ficticio del Coordinador pero que, al mismo tiempo, este no se transformara en un intermediario sobre el cual habría que depositar confianza.

Entre las alternativas se decidió por *Oraclize*, que a través de sus "pruebas de autenticidad" basadas en máquinas virtuales auditables, permite que cualquier *smart contract* acceda a datos de API o sitios web de manera segura y descentralizada. Según la página oficial [42], dependiendo de las necesidades del desarrollador, las "pruebas de autenticidad" pueden ser de cuatro tipos: *TLSNotary Proof*, *Android Proof*, *Ledger Proof* y *Storage and Delivery*. Cada una de estas pruebas se puede invocar desde un *smart contract* para corroborar que la información extraída desde una base de datos externa a una *blockchain* es fidedigna. Además, junto a la "prueba de autenticidad", *Oraclize* proporciona una "prueba de almacenamiento" (*proofStorage*) que se ocupa de cargar y guardar la devolución de datos solicitados en IPFS, que es un protocolo y sistema de archivos tipo *peer to peer* que sustituye los protocolos HTTP tradicionales por una web distribuida [43]. Básicamente, para incorporar la herramienta *Oraclize* al sistema de trazabilidad de la

Figura 5.6, fue necesario instalar, dentro de la *DApp*, un *contrato* adicional titulado "`usingOraclize.sol`", con el fin de invocarlo dentro del código del *smart contract* (condición excluyente) [42]. Con ello presente, se grabó la consulta de manera explícita dentro de "`EnergyTraceability.sol`" mediante las funciones "`oraclize_query()`" para efectuar la solicitud y "`callback()`" para recibirla (como se verá en el ítem 5.2.2). De este modo, para un estado cualquiera del SEN reducido, el *smart contract* ejecutará las líneas de código y, conforme *Oraclize* efectúe las consultas programadas, devolverá los datos certificados por *Sello Sol* (*Blockchain GTIME*) y los del servidor ficticio del Coordinador en tiempo real.

La última pieza de la *DApp* es el archivo *JavaScript* titulado "`app.js`", que se ocupa de recibir la información validada por el *smart contract* y de aplicar el Método de Estampillado Postal, así como de la publicación de la *cadena de suministro* en la dirección IP de "loopback" (como se verá en el ítem 5.2.3). La ejecución de este archivo es local, esto quiere decir que los cálculos y la formulación de los resultados gráficos se hacen dentro de la computadora en la que está instalada la *DApp* y, como tal, admite ediciones de código inclusive mientras se ejecuta la aplicación (a diferencia del *smart contract* que, una vez desplegado en *Ethereum*, es inmutable). Para que "`app.js`" opere según su objetivo, este dispone de funciones programadas para "escuchar" la activación de unas variables denominadas "`event`", que como su nombre lo indica, corresponden a sucesos en tiempo real que modifican su valor dentro del *smart contract* dependiendo del estado del sistema observado. Dichas variables se activan según los criterios del desarrollador, pero en este caso se han situado dentro de las funciones "`oraclize_query()`" y "`callback()`" para sensar el envío de consultas web y su posterior respuesta desde los servidores.

A modo de resumen, en sintonía a la *Figura 5.6*, es bueno tener presente que si se dispone de los vectores de energía levantados al *Blockchain GTIME* y de los resultados de la simulación de Flujos Gruesos en los 14 estados horarios grabados en el servidor ficticio del Coordinador (*Tabla 5.9*), es posible solicitar y recibir, desde un *smart contract* desplegado en la red pública de *Ethereum*, los parámetros instantáneos necesarios para prorratear las pérdidas locales de Cerro Navia en el archivo *JavaScript*; desde donde será posible mostrar, a través de la dirección web de "loopback" o "servidor virtual" (*localhost*), la *cadena de suministro* instantánea entre PV Salvador y el cliente final.

En lo que sigue, se presentará el código del *smart contract* (ítem 5.2.2) y algunas partes del archivo *JavaScript* (ítem 5.2.3) como prueba de desarrollo del sistema de trazabilidad, explicando someramente sus componentes más relevantes. Para finalizar, en el ítem 5.2.4, se mostrará de manera breve cómo luce la *DApp* desde la ejecución, explicitando cuáles son los pasos para ello.

5.2.2 Smart Contract – EnergyTraceability.sol

A continuación, se muestra la codificación del *smart contract* desarrollado en el IDE de programación Visual Studio Code (ver *Figura 5.7*). El archivo real puede consultarse y ha sido entregado como material complementario a esta Memoria, el cual puede probarse en cualquier IDE de programación con la *DApp* instalada, o bien, mediante ejecuciones de prueba en el IDE virtual Remix de *Ethereum* [52].

De la codificación del *smart contract* "`EnergyTraceability.sol`" vale la pena destacar que, en lenguaje Solidity, todos los contratos deben iniciarse explicitando la versión utilizada del programa con la frase "`pragma solidity ^'versión'`". Luego, en coherencia a la descripción básica del ítem 5.2.1, se declara el uso de *Oraclize* para realizar consultas fuera de *Ethereum*; para ello, se importa el contrato instalado en la *DApp* y se invoca con la sentencia "`is usingOraclize`" en la apertura del código del contrato.

```
//-------------------------------------ENCABEZADO-----------------------------------------//


pragma solidity ^0.4.25;
import "installed_contracts/oraclize-api/contracts/usingOraclize.sol";

contract EnergyTraceability is usingOraclize {

    address public owner;
    string public vectorEnergy;
    string public vectorLosses;
    bytes32 queryId1;
    bytes32 queryId2;
    mapping(bytes32=>bool) validIds;

    event newOraclizeQuery(string description);
    event LogEnergyUpdate(string energy);
    event LogLossesUpdate(string losses);
    event LogUpdate(address indexed _owner, uint indexed _balance);


//-------------------------------------CONSTRUCTOR----------------------------------------//

    constructor() payable public{

    owner = msg.sender;
    emit LogUpdate(owner, address(this).balance);
    oraclize_setCustomGasPrice(4000000000);

    //OAR para Oraclize
    OAR = OraclizeAddrResolverI(0x6f485C8BF6fc43eA212E93BBF8ce046C7f1cb475);

    oraclize_setProof(proofType_TLSNotary | proofStorage_IPFS);
    update();

    }
```

```
//-----------------------------------CONSULTAS DE ORACLIZE---------------------------------//

    //Función Fallback
    function() public{
        revert();
    }

    function __callback(bytes32 myid, string result, bytes proof) public {
        if (!validIds[myid]) revert();
        if (msg.sender != oraclize_cbAddress()) revert();

        if (myid == queryId1){
        vectorEnergy = result; //recibe datos de Blockchain GTIME en variable tipo “string”
        emit LogEnergyUpdate(vectorEnergy);
        delete validIds[myid];
        update();
        }

        if (myid == queryId2){
        vectorLosses = result; //recibe información del Coordinador en variable tipo "string"
        emit LogLossesUpdate(vectorLosses);
        delete validIds[myid];
        update();
        }

    }

    function update() payable public {
    emit newOraclizeQuery("Oraclize - Sello Sol Query");

//Solicitud de datos de inyección y retiro a Blockchain GTIME con XPath JSON para respuesta
    queryId1 = oraclize_query("URL",
"json(http://45.7.229.137/api/supplyChain).response.[eGen,eDem]",
'{"genId":"94f2e7668dcf7b45c6291881b6bf2c962a20d0f71d52152f7a7e7160fa18e223",
"consId":"bebeee44d2296630f1c39683f0e008b9d6099be9770c0d373f95ffc741f76fd9"}',600000);

//Solicitud de datos de pérdidas locales a Coordinador con XPath JSON para respuesta
    queryId2 = oraclize_query("URL",
"json(http://45.7.229.137/api/totalLosses).response.[k,Dgross,Dreal]",400000);

//Se asigna valor booleano a las consultas “queryId” para agregarlas al mapping
    validIds[queryId1] = true;
    validIds[queryId2] = true;
   }

}
```

Figura 5.7: *Smart contract* “`EnergyTraceability.sol`” desarrollado en Visual Studio Code y en Remix. Elaboración propia.

Lo que sigue es la declaración de las variables globales del *smart contract*, para las que se debe definir el tipo y nombre, respectivamente. Así, como se aprecia en la *Figura 5.7*, se definen dos variables tipo “`string`” y dos variables tipo “`bytes32`” (número entero) que serán útiles para realizar las solicitudes al *Server* de *Blockchain GTIME* y al servidor ficticio del Coordinador (ver *Figura 5.6*). Al respecto, la variable tipo “`mapping`” es un arreglo diseñado para guardar en memoria las consultas válidas realizadas por

Oraclize. También, se define una variable tipo "`address`", que es una variable nativa de Solidity útil para diferenciar las *direcciones* de cuenta de los nodos de *Ethereum*, y que, en este caso, está reservada para observar la *clave pública* de quien pondrá en marcha el contrato en la red con *blockchain* ("`owner`").

Luego se puede ver la declaración de cuatro sucesos tipo "`event`", que se activan respectivamente con la sentencia "`emit`" cuando: se ejecuta la función "`update()`", se ejecuta la función "`callback()`" para consultas realizadas según "`queryId1`", se ejecuta la función "`callback()`" para consultas realizadas según "`queryId2`" y cuando se despliega el *smart contract* (ver *Figura 5.7*).

Cuando el "`owner`" del *smart contract* despliega el archivo en la *blockchain pública* de *Ethereum*, se ejecuta el "`constructor`", en donde lo primero que se nota es que se almacena de forma permanente la *dirección* de quien ha creado el contrato en la red gracias a "`msg.sender`". También, en el "`constructor`" se establece el precio que el "`owner`" del contrato está dispuesto a pagar para que el archivo trabaje, con la línea "`oraclize_setCustomGasPrice()`". Aquí es bueno comentar que en el mundo de *Ethereum* dicho valor se denomina *precio de gas* y corresponde al pago que se debe hacer para que los nodos de la red *pública* validen las transacciones del *smart contract*. En la práctica, se paga el costo computacional u operacional del nodo validador a fin de perpetuar el funcionamiento del *contrato*. El pago lo efectúa el "`owner`" del archivo mediante WEI, que es el formato contable de cada unidad de *gas* (unidad operacional); siendo la equivalencia entre la criptomoneda de *Ethereum* y el medio de pago de las operaciones en la misma igual a (en Giga-WEIs): 1 ETH = 1.000.000.000 GWEI [46]. Con lo anterior, en el código se definieron 4 GWEI como monto a pagar por cada transacción (ver *Figura 5.7*).

Una última observación sobre el "`constructor`" la constituye la declaración de la *dirección* del contrato de *Oraclize* y la enunciación de la prueba de autenticidad "`proofType_TLSNotary`" y de almacenamiento "`proofStorage_IPFS`" con las cuales se recurrirá descentralizadamente a las fuentes de datos convenidas.

En la sección final del código, están las "consultas de *Oraclize*", donde se exhibe la función "`update()`" que elabora las preguntas y la función "`callback()`" que procesa las respuestas (ver *Figura 5.7*). Para la primera de ellas, luego de notificar la ocurrencia de su ejecución (mediante el "`emit`" de la variable correspondiente), declara dos tipos de solicitud de datos: "`queryId1`" y "`queryId2`"; las que solicitan los datos de energía a la dirección del *Server* de *Blockchain GTIME* y los datos de Flujos Gruesos a la dirección del servidor ficticio del Coordinador, mediante la definición de las fuentes URL con los métodos POST y GET, para cada caso. Además, note que como requisito para consultar los vectores *GTIME*, en "`queryId1`" se debe proporcionar las *ID* de los medidores *phiNet* y *phiNergy*, explicitando que se desea conocer las cuantías de energía de producción "`eGen`" y demanda "`eDem`", respectivamente. Para el caso de

“queryId2”, en tanto, basta con entregar los parámetros que se desea consultar, como los resultados “k”, “Dgross” y “Dreal” (ver *Tabla 5.9*). Bajo cualquier caso, cuando “update()” corre, se registra en el “mapping” el ‘request’ realizado. De la función “callback()” vale la pena destacar que lo primero que hace es comprobar si la consulta que se efectuó está almacenada en el arreglo “validIds” del *smart contract*, así como si esta proviene del contrato de *Oraclize* invocado por el “owner” de “EnergyTraceability.sol”. Si se confirma lo anterior, la función somete la respuesta (esto es, los datos importados agrupados en formato “string” en la variable “result”) a una prueba lógica, a fin de discriminar si la información recibida proviene de la plataforma *Sello Sol* (basada en *Blockchain GTIME*) o del servidor ficticio del Coordinador (ver *Tabla 5.9*). Cuando se tiene claridad de lo anterior, la función “callback()” activa el evento correspondiente y, con ello, puede ceder los datos al archivo *JavaScript* para que efectúe los cálculos de Prorrateo y de construcción visual de la *cadena de suministro*.

5.2.3 Archivo JavaScript – App.js

En este caso se describirá someramente el archivo “app.js” de la *DApp*, el cual ha sido extraído desde el blog de la compañía Medium [53], pero que ha sido modificado y adaptado al sistema de trazabilidad de energía perseguido en este trabajo. El código completo del archivo *JavaScript* está disponible en la carpeta “oraclize-test” ubicada dentro de la *DApp*, que ha sido entregada como material complementario a este documento. Para empezar, en la *Figura 5.8*, se muestra una de las primeras secciones del archivo “app.js”, en ella se encuentra codificada la importación del código del *smart contract*, pero en un formato de texto descifrable por el lenguaje *JavaScript*, denominado “.json”; permitiéndole al archivo “app.js” interpretar correctamente los parámetros entregados por “EnergyTraceability.sol”.

```
// Importa los "artefacts" del smart contract y los convierte en abstracciones utilizables.
import contract_build_artifacts from '../../build/contracts/EnergyTraceability.json'

//ETContract es nuestra abstracción utilizable, que usaremos a través del código a
continuación.
var ETContract = contract(contract_build_artifacts);
var accounts;         var account;
var K_G2;             var D_gross;
var D_real;           var G_2;
var D_cliente;        var G_2_11;
var P_losses_11;      var P_losses_11_2;
var D_real_11_2;      var K_D11;
var G_cliente;        var P_losses_cliente;
```

Figura 5.8: Encabezado archivo *JavaScript* “app.js” adaptado de [53] y desarrollado en Visual Studio Code. Elaboración propia.

También, de la imagen anterior se aprecia la declaración de las variables que heredarán las cuantías consultadas por *Oraclize* a los servidores de información (con la palabra reservada "`var`"), y que, mediante su correcta manipulación, permitirán aplicar el Método de Estampillado Postal de manera análoga al ejemplo realizado en la sección 5.1.3.3.

```
// Abre un "socket" y "escucha" los eventos definidos en nuestro contrato
  addEventListeners: function(instance){

    var newOraclizeQuery = instance.newOraclizeQuery({},{fromBlock: 0, toBlock: 'latest'});
    var LogEnergyUpdate = instance.LogEnergyUpdate({},{fromBlock: 0, toBlock: 'latest'});
    var LogLossesUpdate = instance.LogLossesUpdate({},{fromBlock: 0, toBlock: 'latest'});
    var LogCreated = instance.LogUpdate({},{fromBlock: 0, toBlock: 'latest'});
```

Figura 5.9: Método "`addEventListeners()`" en "`app.js`" codificada en Visual Studio Code. Elaboración propia.

Por otra parte, en la *Figura 5.9* se exhibe el método que actúa como "sensor" de los cambios de estado de las variables tipo "`event`" del *smart contract*. En otras palabras, la función "`addEventListeners()`" está atenta a la ocurrencia de consultas efectuadas por *Oraclize* mediante la observación de la *cadena de bloques* construida por los nodos de *Ethereum*. Para ello, la función reúne un conjunto de cuatro métodos: uno por cada evento del *smart contract* (ver *Figura 5.7*).

En relación con lo anterior, la *Figura 5.10* expone la lógica de actuación de dos de los métodos encargados de "escuchar" los cambios de estado del "`event`" para el cual fueron diseñados. Al respecto, el método "`LogEnergyUpdate.watch()`" toma los datos entregados por la función "`callback()`" del *smart contract* mediante una única variable tipo "`string`" llamada "`result`". Luego, con la aplicación de la función "`JSON.parse()`" se consigue una parcelación legible del contenido, con la cual *JavaScript* puede trabajar. Evidencia de ello es la creación de un arreglo llamado "`array_energy`" cuyos elementos son la energía instantánea generada por PV Salvador y consumida por el cliente final procedentes del *Server* de *Blockchain GTIME*, grabadas en las variables "`App.ethEnergyG`" y "`App.ethEnergyD`", respectivamente.

Observe que con el método "`LogLossesUpdate.watch()`" sucede algo similar, ya que este recibe la variable "`result`" entregada por la función "`callback()`" del *smart contract*, para después aplicar la función "`JSON.parse()`" y obtener la creación de un arreglo de tres elementos llamado "`array_losses`". El contenido de este vector es el factor de contribución de producción energética de PV Salvador k_{G_2}, la *demanda gruesa* $\mathcal{D}_{11}$ y la *demanda real* $\mathcal{D}_{11}^{real}$ (ver *Tabla 5.9*); y sus cuantías se han separado en las variables "`App.ethK`", "`App.ethDgross`" y "`App.ethDreal`", en cada caso (ver *Figura 5.10*).

```
LogEnergyUpdate.watch(function(err, result){

  console.log('control');
  if(!err){
    //Muestra energía en formato "string" según evento "LogEnergyUpdate"
    console.log('********************', (result.args.energy))

    //transforma la propiedad "energy" del objeto "result" de tipo "string" a "JSON"
    let array_energy=JSON.parse(result.args.energy);
    App.ethEnergyG = array_energy[0];//asigna el elemento [0] de "array_energy"
    App.ethEnergyD = array_energy[1];//asigna el elemento [1] de "array_energy"

  }
  else{
    console.log(err)
  }
})


LogLossesUpdate.watch(function(err, result){

  console.log('quiz');
  if(!err){
     //Muestra la energía en formato "string" según evento "LogLossesUpdate"
    console.log('++++++++++++++++++++++', (result.args.losses))

    //transforma la propiedad "energy" del objeto "result" de tipo "string" a "JSON"


    let array_losses=JSON.parse(result.args.losses);
    App.ethK = array_losses[0];//asigna el elemento [0] de "array_losses"
    App.ethDgross = array_losses[1];//asigna el elemento [1] de "array_losses"
    App.ethDreal = array_losses[2];//asigna el elemento [2] de "array_losses"

 //se ejecuta función "showBalance" con los cinco parámetros de entrada ya definidos
 App.showBalance(App.ethK, App.ethDgross, App.ethDreal, App.ethEnergyG, App.ethEnergyD);

  }
  else{
    console.log(err)
  }
})
```

Figura 5.10: Conversión del formato de los datos entregados por el *smart contract* en métodos del "`app.js`". Elaboración propia.

Lo que resta de la imagen anterior es la instrucción "`App.showBalance()`" que se usa para ejecutar la función "`showBalance()`" mostrada en la *Figura 5.11*, tomando como datos de entrada los valores importados por *Oraclize* en formato "`.json`", a saber: k_{G_2}, $\mathcal{D}_{11}$, $\mathcal{D}_{11}^{real}$, G_2 y $\mathcal{D}_{Cliente}$.

Respecto de la función "`showBalance()`", se debe acentuar que esta se ocupa, en parte, de actualizar la *cadena de suministro* que se muestra en el *localhost* de la computadora que contiene la *DApp*. Esto lo logra

usando una *Webpack* que es un “bundler” utilizado por los desarrolladores de aplicaciones web que permite actualizar la interfaz vista por el usuario ayudándose de archivos *CSS* y *HTML* [53].

```
showBalance: function(k, Dgross, Dreal, energyG, energyD){

    //Balance actualizado, inicia animación CSS
    var row = document.getElementById('row');
    row.style.animation = 'heartbeat 0.75s';

    //Elimina la animación CSS después de 1100 ms
    setTimeout(function(row){
      var row = document.getElementById('row');
      row.style.animation = null;
    }, 1100)


    //Inputs para Cerro Navia (barra 11). Se redondea la cuantía racional a tres decimales
    var KG2 = document.getElementById("k");
    K_G2 = parseFloat(k).toFixed(3);
    KG2.innerHTML = K_G2; //Factor de contribución en generación: PV Salvador –> Cerro Navia

    var Dem_gross_11 = document.getElementById("Dgross");
    D_gross = parseFloat(Dgross).toFixed(3);
    Dem_gross_11.innerHTML = D_gross; //Demanda gruesa en Cerro Navia

    var Dem_real_11 = document.getElementById("Dreal");
    D_real = parseFloat(Dreal).toFixed(3);
    Dem_real_11.innerHTML = D_real; //Demanda real en Cerro Navia

    var Gen_2 = document.getElementById("energyG");
    G_2 = parseFloat(energyG).toFixed(3);
    Gen_2.innerHTML = G_2; //Inyección instantánea PV Salvador

    var Dem_cliente = document.getElementById("energyD");
    D_cliente = parseFloat(energyD).toFixed(3);
    Dem_cliente.innerHTML = D_cliente; //Retiro instantáneo Cliente


    //Cálculos sobre Cerro Navia (barra 11). Se redondea la cuantía racional a tres decimales

    //Datos necesarios para Cadena de Suministro a nivel de transmisión nacional
    var Gen_2_11 = document.getElementById("G_2-11");
    G_2_11 = (K_G2 * G_2).toFixed(3);
    Gen_2_11.innerHTML = G_2_11; //Contribución de PV Salvador a Cerro Navia

    var Plosses_11 = document.getElementById("Plosses11");
    P_losses_11 = (D_gross - D_real).toFixed(3);
    Plosses_11.innerHTML = P_losses_11; //Pérdidas locales en Cerro Navia según Flujos Gruesos

    var Plosses_11_2 = document.getElementById("Plosses11-2");
    P_losses_11_2 = ((G_2_11 / D_gross) * P_losses_11).toFixed(3);
    Plosses_11_2.innerHTML = P_losses_11_2; //Pérdidas de abastecimiento desde PV Salvador

    var Dreal_11_2 = document.getElementById("Dreal11-2");
    D_real_11_2 = (G_2-11 – P_losses_11_2).toFixed(3);
    Dreal_11_2.innerHTML = D_real_11_2; //Demanda en Cerro Navia alimentada por PV Salvador
```

```
    //Datos necesarios para Cadena de Suministro a nivel de transmisión zonal y distribución
    var KD11 = document.getElementById("k_D11");
    K_D11 = parseFloat(D_cliente / D_real_11_2).toFixed(3);
    KD11.innerHTML = K_D11; //Factor de contribución en consumo: Cliente -> Cerro Navia

    var Gen_Cliente = document.getElementById("GCliente");
    G_cliente = (K_D11 * G_2_11).toFixed(3);
    Gen_Cliente.innerHTML = G_cliente; //Contribución de PV Salvador al Cliente

    var Plosses_Cliente = document.getElementById("PlossesCliente");
    P_losses_cliente = (K_D11 * P_losses_11_2).toFixed(3);
    Plosses_Cliente.innerHTML = P_losses_cliente; //Pérdidas prorrateadas al Cliente

  }
}
```

Figura 5.11: Función "`showBalance()`" inmersa en el archivo *JavaScript* "`app.js`" de la *DApp*. Elaboración propia.

Además de lo anterior, como se exhibe en la *Figura 5.11*, la función "`showBalance()`" se encarga de efectuar los cálculos de las pérdidas eléctricas que se asignan al cliente final bajo los criterios del Método de Prorrateo estudiados a lo largo de este capítulo. Note que la imagen anterior evidencia la transformación de las variables en formato "`.json`" a una cuantía racional mediante el uso del método "`parseFloat()`". También se expone el uso implícito de la *Ecuación 5.1* gracias a la ponderación de la producción energética de PV Salvador con la variable "`G_2_11`", al cálculo de las pérdidas locales en Cerro Navia con "`P_losses_11`" y al despeje de cobertura máxima con "`D_real_11_2`". Luego, en función de la *Ecuación 5.2* y gracias a la proporción de consumo del cliente final "`K_D11`", se pueden conocer las variables "`G_cliente`", "`P_losses_cliente`" y "`D_cliente`", quienes terminan por conformar la cadena *de suministro* instantánea de electricidad a nivel del consumidor.

Finalmente, mediante el sufijo "`innerHTML`" acoplado a las variables de interés (ver *Figura 5.11*), se consigue el despliegue web de las mediciones y resultados a través del *localhost*, que se lleva a cabo por los archivos titulados "`app.css`" e "`index.html`" de la *Webpack*; los que pueden consultarse dentro de la carpeta "app" contenida en la carpeta "oraclize-test" del conjunto de archivos de la *DApp*, entregada como material complementario a esta Memoria.

5.2.4 Ejecución de la DApp

En este ítem, se mostrarán los pasos a seguir para correr la *DApp*, describiendo las acciones y características más importantes dentro del proceso. Conforme a lo descrito en la fuente de referencia [53], para que una *DApp* pueda operar correctamente, conviene tener instalado el "framework" de desarrollo Truffle que, trabajado en "Node.JS" (el mismo lenguaje de comunicación M2M de los nodos de *Blockchain GTIME*; ver Capítulo 3), es el software más avanzado y ampliamente utilizado para compilar, testear y desplegar *smart contracts* [54]. En consideración de lo anterior, el procedimiento de ejecución es como sigue:

1. Una vez abierto el IDE de programación (Visual Studio Code en este caso), diríjase a la carpeta "ethereum-training".

2. Abra el terminal (Windows PowerShell, en este caso) y dentro de la carpeta "oraclize-test" ejecute el comando "`truffle-develop`". Con esta instrucción se cargará el conjunto de cuentas de *Ethereum* que, a nivel de prototipo, simularán la red *pública* distribuida con la cual se validarán las transacciones desde y hacia el *smart contract* "`EnergyTraceability.sol`". Lo anterior se hace porque, dada la inmutabilidad de una red con *blockchain*, es bueno usar redes de prueba como "TestRPC" (usada en este caso) o "Ganache" para testear y comprobar la funcionalidad de los *contratos inteligentes* antes de desplegarlos realmente en la web (ver *Figura 5.12*).

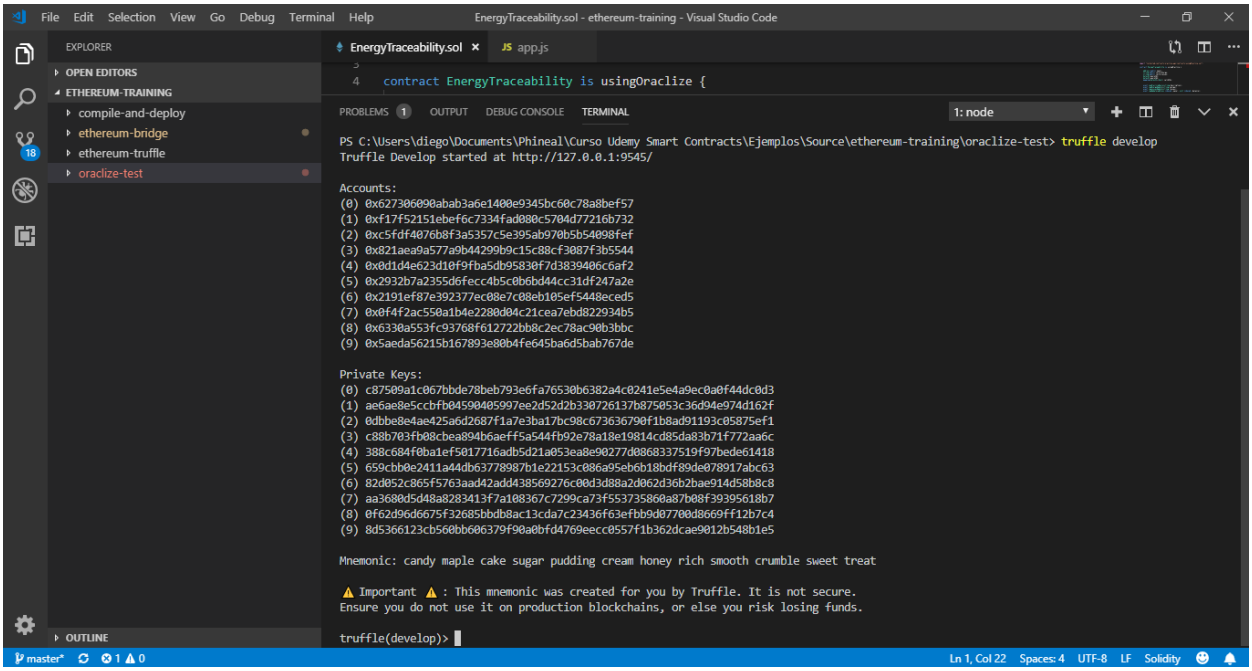


Figura 5.12: *Dirección* y *clave privada* de los 10 nodos (cuentas) que conforman la red de prueba "TestRPC". Elaboración propia.

3. Abra un nuevo terminal y, dentro de la carpeta “ethereum-bridge” ingrese el comando “`node bridge -a 9 -H 127.0.0.1 -p 9545 --dev`”. Según se explica en [53], esta instrucción ordena a los archivos de la carpeta “ethereum-bridge” a usar la última cuenta creada en la red de prueba (nodo 9, según “`-a 9`”, ver *Figura 5.12*) para crear el contrato “`usingOraclize.sol`” de *Oraclize*. Por su parte, con “`-H 127.0.0.1`” se establece la dirección *IP* del *localhost* y con “`-p 9545`” se designa el puerto con el que se identificará el proceso dentro de la computadora que ejecuta la *DApp*. A nivel de la interfaz esto es importante, ya que con ello el navegador web “ubica” la red prueba de *Ethereum*, permitiendo mostrar la evolución de las salidas de la *DApp* a través del tiempo.

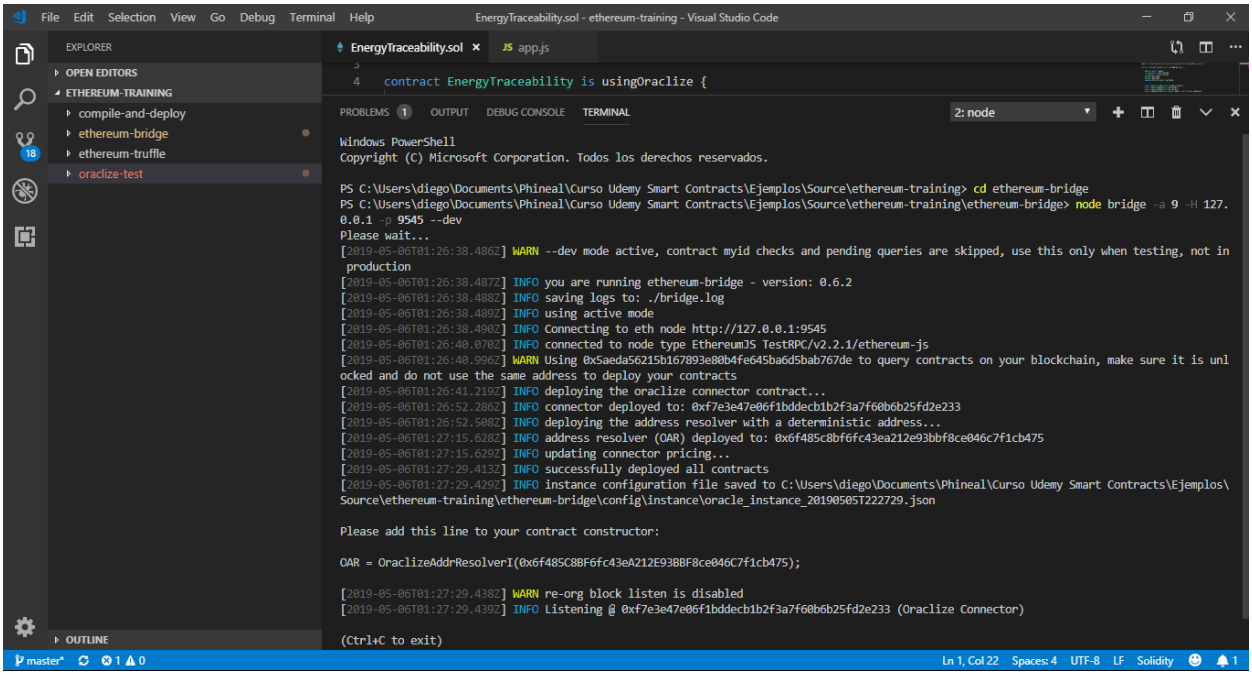


Figura 5.13: Designación de cuenta, dirección de “loopback” y de puerto para la red de prueba “TestRPC”. Se observa la obtención de la *dirección* del contrato de *Oraclize* (`OAR`) que Truffle recomienda declarar en “`EnergyTraceability.sol`”. Elaboración propia.

4. Abra un nuevo terminal y diríjase a la carpeta “oraclize-test”. Una vez allí, ingrese el comando “`truffle-compile`” para compilar el *smart contract*. Luego, escriba el código “`truffle migrate --development --reset`” para desplegar el contrato “`EnergyTraceability.sol`” dentro de “TestRPC”. Con esto, el *smart contract* adquiere una *dirección* en la red ficticia de *Ethereum*, con la cual el resto de los nodos puede interactuar (ver *Figura 5.14*). Finalmente, para que la aplicación corra y utilice la *Webpack* prevista, escriba “`npm run dev`” y se accionará el “`webpack-dev-server`” de Truffle, con el que se muestra la *cadena de suministro* instantánea en el navegador web, exhibiendo las cuantías de inyección, retiro y pérdidas (ver *Figura 5.15*).

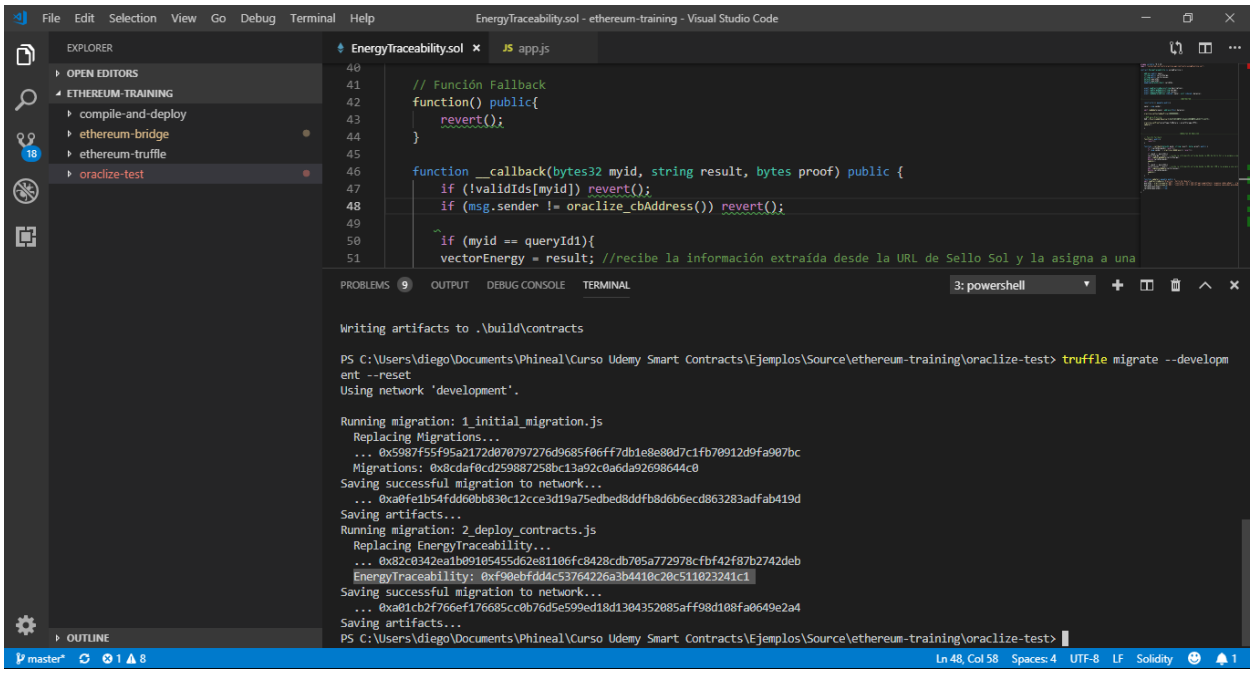


Figura 5.14: Despliegue de *smart contract* en "TestRPC". Se aprecia que la *dirección* de "`EnergyTraceability.sol`" dentro de la red de prueba de *Etehreum* es, en este caso: "`0xf90ebfdd4c53764226a3b4410c20c511023241c1`". Elaboración propia.

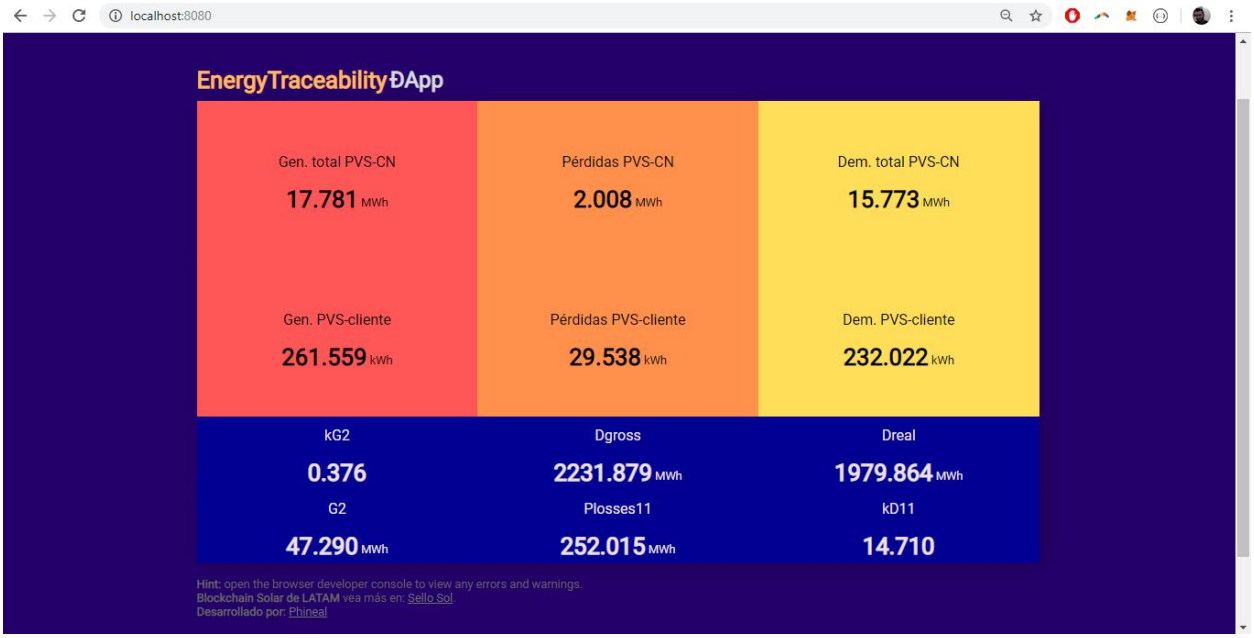


Figura 5.15: Aspecto gráfico del despliegue de la *DApp* en la dirección de "loopback" del navegador web para el 15/05/2019 a las 16:00 hrs. Al igual que a lo largo de este documento, se decidió mantener la coherencia en los colores utilizados para producción, transporte y consumo. Se distingue la *cadena de suministro instantánea* a nivel de transmisión nacional (arriba) y a nivel del cliente final (abajo), así como los parámetros más relevantes para su construcción (en el cuadro azul). Para efectos comerciales, la exposición de dichos parámetros instantáneos, que surgen de la aplicación del Método de Bialek y del Método de Estampillado Postal, puede obviarse, ya que sólo cumplen un rol demostrativo. Elaboración propia.

6. Conclusiones

Con el propósito de validar un sistema de trazabilidad de energía eléctrica basado en *blockchain*, se estudiaron diversos mecanismos de asignación de pérdidas en transmisión a fin de concretar una propuesta metodológica coherente a los objetivos de este trabajo. Al respecto, con el Método de Bialek aplicado sobre un modelo simplificado del Sistema Eléctrico Nacional (SEN) y con el Método de Estampillado Postal aplicado sobre la barra Cerro Navia de dicho modelo, fue posible conseguir la construcción teórica de una *cadena de suministro* de electricidad a nivel del cliente final, asumiendo las pérdidas incurridas a nivel nacional, capaz de mutar en cada estado del SEN reducido y con la facultad de implementarse sobre un prototipo tecnológico del sistema.

Luego, con el afán de concretar empíricamente una *cadena de suministro* instantánea de electricidad a nivel del cliente final, se programó un *smart contract* que, desplegado sobre una red de prueba de *Ethereum*, sirve para gestionar tanto los datos de energía procedentes desde el *Server* de *Blockchain GTIME* (mediciones en tiempo real) como los parámetros para el cálculo de pérdidas a nivel nacional provenientes desde el servidor ficticio del Coordinador (resultados de la aplicación de Flujos Gruesos en los 14 estados horarios del SEN reducido). Con ello, el *smart contract*, formando parte de una *aplicación descentralizada* (*DApp*), hace de intermediario entre las bases de datos y los elementos que conforman la *DApp*. Entre todos, el archivo "`app.js`" es quien se encarga de recibir los datos de energía gestionados por el *smart contract* para posteriormente ajustar las cuantías de generación y pérdidas conforme al Método de Prorrateo a nivel del cliente final; lo que termina por constituir la trazabilidad completa de los flujos, esto es, desde la producción al consumo.

Con todo lo anterior, se pretende enseñar que a lo largo de este documento se evidenció el modelo propuesto, las variables y las condiciones necesarias para garantizar la proveniencia de la electricidad consumida por un cliente final; desde la óptica de la justificación metodológica hasta los atributos de trazabilidad implícitos en la herramienta tecnológica utilizada. De este modo, se obtiene un refuerzo importante para certificaciones como las de la iniciativa *Sello Sol*, que busca dotar de una distinción comprobable, segura e inmutable a los productos y servicios llevados a cabo con energía solar; permitiendo que los usuarios accedan a la *cadena de suministro* empleada para ello, la cual se construye y audita sin intervención humana.

También, se ha dejado "la puerta abierta" para probar esquemas de tarificación a través del desarrollo de nuevos contratos de abastecimiento llamados *Smart PPA*, en los que tanto el productor como el consumidor pueden estar al tanto del cumplimiento de las cantidades acordadas en tiempo real, definiendo un precio para valorizar la energía. Otra aplicación potencial la constituyen mecanismos

de trazabilidad de mitigación de emisiones de CO2, en donde la *cadena de suministro* no sólo mostraría el efecto de las plantas "limpias", sino que además se estaría al tanto del grupo de "consumidores verdes" o con baja huella de carbono.

Por otra parte, se debe advertir que el desarrollo del proyecto está en fase de prototipo, por lo que es susceptible a mejoras, adaptaciones y reformas. Entre ellas hay espacio para elaborar un modelo más adecuado del SEN (quizás mediante un acuerdo de colaboración con el Coordinador u otro organismo técnico responsable, quien provea un modelo reducido oficial y que otorgue acceso a los equipos de telemetría más relevantes), automatizar la aplicación del Método de Bialek, mejorar la frecuencia de muestreo de datos (ya sea del *Blockchain GTIME* como del servidor del Coordinador), mejorar la interfaz gráfica, levantar la información a la web, dar acceso al rastreo por parte de usuarios, refutar o perfeccionar la propuesta metodológica, etc.

Para cerrar, y de manera adicional al cumplimiento de los objetivos formales, también es bueno compartir que la elaboración de este trabajo representa una muestra del diálogo que se está dando entre las tecnologías de generación y de almacenamiento (como aquellas basadas en energías renovables) con las de información (como *blockchain* aplicado al sector Energía). Al respecto, resulta interesante notar que, por ejemplo, la modularidad de las instalaciones fotovoltaicas y de almacenamiento tienen el potencial de favorecer la creación de nuevos puntos de inyección en clientes finales a nivel de transmisión zonal y distribución, relegando el uso de las redes como meros medios de transporte y adoptando tecnologías de gestión de datos más inteligentes, seguras y transparentes.

Lo anterior implica el reemplazo de los intermediarios convencionales por una interacción descentralizada entre los usuarios de la red, en donde el uso y comercio de electricidad quedaría en manos de los prosumidores. Quizás el destino más próximo para la adopción de estos mecanismos de la llamada "revolución industrial 4.0" no sea en grandes sistemas eléctricos, sino que en "microgrids" y en comunidades de energía colaborativa. Proyectos como *LO3 Energy* (citado en el Capítulo 2) demuestran que en sistemas reducidos es posible reproducir nuevos paradigmas de intercambio de electricidad a través de *blockchain*, exhibiendo el *pool* de generadores en tiempo real y dando espacio a los consumidores para escoger la procedencia de su energía. En sistemas pequeños pero más complejos (como una comuna, por ejemplo), la incorporación de metodologías que permitan conocer la *cadena de suministro* dentro de la red puede significar avances importantes para dar con la responsabilidad de los agentes en el uso del sistema de transporte y, con ello, conseguir un sistema colaborativo de energía más completo y económicamente sustentable.

7. Referencias

[1] J. Bialek, “Tracing the flow of electricity”, *IEEE Proceedings on Generation, Transmission and Distribution*, Vol. 143, N°4, pp 313-320, Julio 1996.

[2] D. Kirschen, R. Allan y G. Strbac, “Contributions of individual generators to loads and flows”, *IEEE Transactions on Power Systems*, Vol.12, N°1, Febrero 1997.

[3] M. Chamba, “Propuesta para la eliminación de los factores de nodo en la evaluación de las transacciones comerciales de energía en el mercado eléctrico ecuatoriano”, Proyecto previo a la obtención del título de Ingeniero Eléctrico, Escuela de Ingeniería, Escuela Politécnica Nacional, Quito, Ecuador, 2007.

[4] F. Danitz, “Métodos de asignación de peajes de los sistemas de transmisión eléctrica según el uso de la red”, Memoria para optar al título de Ingeniero Civil de Industrias con mención en Electricidad, Escuela de Ingeniería, Pontificia Universidad Católica de Chile, Santiago de Chile, Chile, 2001.

[5] S. Nakamoto. (2008). *Bitcoin: A peer-to-peer electronic cash system*. Bitcoin. [Online]. Disponible en: `https://bitcoin.org/bitcoin.pdf`

[6] K. Christidis, M. Devetsikiotis, “Blockchains and smart contracts for the internet of things”, *IEEE Access, The Plethora of Research in Internet of Things*, Vol. 4, pp 2292-2303, Mayo 2016.

[7] J. Todó, “Análisis de viabilidad de una microgrid con tecnología blockchain y fuente de energía fotosintética: greencoin”, Trabajo final de Máster Universitario en Ingeniería Industrial, Escola Tècnica Superior d’Enginyeria Industrial de Barcelona, Universitat Politècnica de Catalunya, Barcelona, Cataluña, España, 2017.

[8] S/A, “La cadena de bloques (blockchain): una tecnología disruptiva con el poder de revolucionar el sector financiero”, EquiSoft Inc., EE.UU., Informe Técnico, Marzo 2017.

[9] H. Van Steenis, B. Graseck, F. Simpson, J. Faucette, “Global Insight: Blockchain in banking: disruptive threat or tool”, Morgan Stanley & Co., Morgan Stanley Research, EE.UU., Global Financial/FinTech, Abril 2016.

[10] S. Brakeville, B. Perepa, “Blockchain basics: Introduction to distributed ledgers”, International Business Machines Corporation, EE.UU., IBM Developer, Mayo 2016.

[11] S/A, “Tecnología blockchain. El avance de Bitcoin y de los pagos virtuales”, Banco Bilbao Vizcaya Argentaria , Centro de innovación BBVA/Innovation Edge, España, Serie Fintech, Enero 2016.

[12] Ó. Lage. (2017, Julio 05). *¿Es ‘blockchain’ realmente inmutable?*. BBVA. [Online]. Disponible en: `https://www.bbva.com/es/blockchain-realmente-inmutable/`

[13] S. Chintakrindi. (2017, Junio 21). *Understanding crypto-economic security through Game Theory*. Blockchain at Berkeley. [Online].
Disponible en:
https://blockchainatberkeley.blog/understanding-crypto-economic-security-through-game-theory-526e810c7736

[14] F. Kunze. (2017, Octubre 12). *Introduction to blockchain*. Blockchain at Berkeley. [Online].
Disponible en: https://blockchainatberkeley.blog/introducci%C3%B3n-a-blockchain-658fdf4e0ff3

[15] T. Dinh, R. Liu, M. Zhang, G. Chen, B. Ooi, J. Wang, "Untangling blockchain: A data processing view of blockchain systems", *IEEE Transactions on Knowledge and Data Engineering*, Vol.30, pp 1366-1385, N°7, Julio 2018.

[16] T. Fernández, P. Fraga, "A review on the use of blockchain for the Internet of Things", *IEEE Access*, Vol.6, pp 32979-33001, Mayo 2018.

[17] M. Allende, V. Colina. (2018, Junio 28). *¿Pública, federada o privada? Explora los distintos tipos de blockchain.* Banco Interamericano de Desarrollo. [Online].
Disponible en: https://blogs.iadb.org/abierto-al-publico/2018/06/28/tipos-de-blockchain/

[18] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, V. Santamaría, "To blockchain or not to blockchain: that is the question", *IEEE Computer Society - IT Professional*, Vol.20, pp 62-74, N°2, Marzo/Abril 2018.

[19] Sh. Lee. (2018, Abril 19). *Bitcoin's energy consumption can power an entire country, but EOS is trying to fix that*. Forbes Magazine. [Online].
Disponible en: https://www.forbes.com/sites/shermanlee/2018/04/19/bitcoins-energy-consumption-can-power-an-entire-country-but-eos-is-trying-to-fix-that/#4d68308c1bc8

[20] J. Schneider, A. Blostein, B. Lee, S. Kent, I. Groer, E. Beardsley, "Case Study 2: Building and distributed smart grid with blockchain", Goldman Sachs Group Inc., Goldman Sachs Global Investment Research, EE.UU, Profiles in Innovation: Blockchain, putting theory into practice, Mayo 2016.

[21] S/A (2018, Febrero 26). *Comisión Nacional de Energía se convertirá en la primera entidad pública en utilizar blockchain en Chile*. Comisión Nacional de Energía. [Online].
Disponible en:
https://www.cne.cl/prensa/prensa-2018/febrero-2018/comision-nacional-de-energia-se-convertira-en-la-primera-entidad-publica-en-utilizar-blockchain-en-chile/

[22] S/A (2018, Junio 18). *Bolsa de Santiago lanza su primera aplicación blockchain con nueva versión del sistema de ventas cortas y préstamo de valores*. Bolsa de Comercio de Santiago. [Online].
Disponible en: http://www.bolsadesantiago.com/noticias/Paginas/Bolsa-de-Santiago-lanza-su-primera-aplicaci%C3%B3n-Blockchain-con-nueva-versi%C3%B3n-del-sistema-de-ventas-cortas-y-pr%C3%A9stamo-de-valor.aspx

[23] Phineal. Ingeniería Eléctrica y Energías Renovables. [Online].
Disponible en: http://phineal.com/es/

[24] Phineal, "Sello Sol: El blockchain solar de Latinoamérica para los nuevos mercados energéticos", Sello Sol, Santiago de Chile, Chile, Presentación Sello Sol, v. 1207, Junio 2018.

[25] Phineal. Sello Sol. El blockchain solar de Latinoamérica. [Online].
Disponible en: http://www.sellosol.com/es/inicio/

[26] Phineal. PhiNet. GTIME - El Blockchain de Energía. [Online].
Disponible en: https://www.phinet.cl/es/gtime/

[27] F. Orecchini, A. Santiangeli, F. Zuccari, A. Dell'Era, "The concept of energy traceability: application to EV electricity charging by RES" en ATI2015 - 70th Conference of ATI Engineering Association, Roma, Italia, 2015, pp. 637-644.

[28] L. Pinto (2018, Junio 06). *Trazabilidad en la producción*. Asociación de la Industria Eléctrica y Electrónica de Chile. [Online].
Disponible en: http://www.aie.cl/files/file/comites/ca/articulos/junio-06.pdf

[29] A. Cañigueral, "El futuro nunca estuvo tan presente" en *Ya nada será lo mismo. Los efectos del cambio tecnológico en la política, los partidos y el activismo juvenil*. Barcelona, Cataluña, España: Centro Reina Sofía sobre Adolescencia y Juventud, Fundación de Ayuda contra la Drogadicción, 2015, pp. 311-322.

[30] Slock.it. Share & Charge. [Online].
Disponible en: http://shareandcharge.com/

[31] WePower, "WePower: Energy trading platform powered by blockchain technology", Vilnius, Lituania, White Paper, versión 0.8, Febrero 2018.

[32] Klenergy Tech, "The Pylon Network", Pylon Network, Benicarló, Castellón, Comunidad Valenciana, España, White Paper, v. 170730, Julio 2017.

[33] Greeneum Network, "Greeneum: Global energy trading networks", S/U, White Paper, S/v, Noviembre 2017.

[34] LO3 Energy. The Brooklyn Microgrid. [Online].
Disponible en: https://www.brooklyn.energy/

[35] Raspberry Pi Foundation. [Online]
Disponible en: https://www.raspberrypi.org/

[36] S/A (2015, Septiembre 30). *¿Qué es y para qué sirve Node.JS?*. Blog de NetConsulting. [Online].
Disponible en: https://www.netconsulting.es/blog/nodejs/

[37] Ministerio de Energía, "Establece un nuevo sistema de transmisión eléctrica y crea un organismo coordinador independiente del sistema eléctrico nacional", Ley 20.936, Biblioteca del Congreso Nacional, Chile, 2016. [Online]
Disponible en: https://www.leychile.cl/N?i=1092695&f=2017-02-09&p=

[38] Coordinador Eléctrico Nacional. Portal de información pública. [Online].
Disponible en: https://www.coordinador.cl/sistema-informacion-publica/

[39] Comisión Nacional de Energía, "Complementa y modifica Resolución Exenta N°778, que establece plazos, requisitos y condiciones para la fijación de precios de nudo promedio, de fecha 15 de noviembre de 2016, modificada por Resolución Exenta N°203, de fecha 25 de abril de 2017", Ministerio de Energía, Santiago de Chile, Chile, Resolución Exenta N°558, 2017.

[40] H. Rudnick, "Nueva Ley de Transmisión y Operación del Sistema Eléctrico", presentado ante la Comisión de Minería y Energía, Cámara de Diputados, Valparaíso, Chile, 2015. [Online].
Disponible en:
https://www.camara.cl/pdf.aspx?prmID=38360&prmTIPO=DOCUMENTOCOMISION

[41] Ethereum (07 marzo, 2015). Publicación oficial de Ethereum vía Google+: "Ethereum Gas Prices for Frontier release". [Online].
Disponible en: https://plus.google.com/112222107871954080534/posts/VsGxrewXspC

[42] Oraclize. Documentación de Oraclize. [Online].
Disponible en: https://docs.oraclize.it/#home

[43] IPFS. IPFS, la web distribuida. [Online].
Disponible en: https://ipfs.io/

[44] Energía Abierta. Comisión Nacional de Energía. [Online].
Disponible en: http://energiaabierta.cl/

[45] D. Kirschen, G. Strbac, "Postage Stamp Method", en Fundamentals of Power System Economics, Wiley, 2004, pág. 232.

[46] Blockchain Luxembourg S.A. Portal de Aprendizaje. [Online].
Disponible en: https://www.blockchain.com/learning-portal/bitcoin-faq

[47] Blockchain Demo. A visual demo of blockchain technology. [Online].
Disponible en: https://blockchaindemo.io

[48] Top Mining Profit. Technical specifications for mining and its profitability. Miner AMD Rig 6 Vga RX470. [Online].
Disponible en: https://topminingprofit.com/miners/171++Rig+6+Vga+RX470

[49] Aaron Hankin (2018, Diciembre 17). *Bitcoin peaked one year ago: Here's a look at 12 months of misery*. Financial News London. [Online].
Disponible en: https://www.fnlondon.com/articles/bitcoin-peaked-one-year-ago-heres-a-look-at-12-months-of-misery-20181217

[50] Digiconomist. Bitcoin energy consumption index. [Online].
Disponible en: https://digiconomist.net/bitcoin-energy-consumption

[51] OpenCloud. [Online].
Disponible en: https://www.opencloud.cl/

[52] Remix. El IDE virtual de Ethereum [Online].
Disponible en: http://remix.ethereum.org

[53] Lander Willems (2018, Mayo 11). *How to create a DApp using Truffle, Oraclize, ethereum-bridge and Webpack*. Medium Corporation [Online]. Disponible en:
https://medium.com/coinmonks/how-to-create-a-dapp-using-truffle-oraclize-ethereum-bridge-and-webpack-9cb84b8f6bcb

[54] Truffle. [Online].
Disponible en: https://truffleframework.com

8. Anexos

8.0 Resumen ejecutivo

Este documento se orienta al desarrollo y la validación de un sistema de trazabilidad de energía eléctrica mediante un *smart contract* basado en tecnología *blockchain*. Se dice que hay un "desarrollo del sistema de trazabilidad" debido a que se programa un *smart contract* que, formando parte de una *aplicación descentralizada* (*DApp*), permite conformar una *cadena de suministro* instantánea de electricidad entre un generador puntual y un consumidor final específico. Al mismo tiempo, se dice que hay una "validación del sistema de trazabilidad" por cuanto se incorporan metodologías de asignación de pérdidas dentro de la misma *DApp* con el fin de que la *cadena de suministro* sea coherente a principios físicos y a supuestos propios de la disciplina eléctrica.

Para entender mejor los temas abordados, vale la pena mencionar que la tecnología *blockchain* o *cadena de bloques* funciona como una base de datos distribuida entre todos los nodos o computadores que conforman una red de información (ver *Figura A.0.1*). El hecho de que la topología de red informática sea distribuida implica que cada nodo de la red tiene una réplica exacta de la base de datos (esto es, una copia de la *blockchain*), a diferencia de lo que ocurre en una red informática tradicional donde la base de datos se aloja en un servidor único o centralizado.

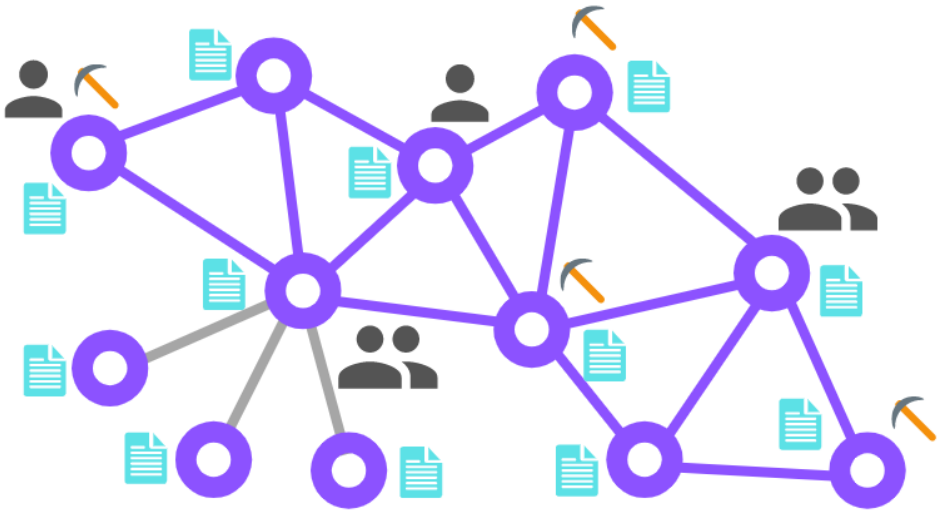


Figura A.0.1: Esquema de una red con *blockchain*. La topología de red de información distribuida permite que todos los nodos tengan una copia de la base de datos (la *blockchain* misma, en celeste). Los nodos interactúan gracias a *firmas digitales* y pueden representar a más de un usuario. Algunos nodos usan la red para efectuar transferencias de *activos* y otros se encargan de aprobar las transacciones según el *protocolo de consenso* programado en la red (nodos *mineros*, simbolizados por una picota). A pesar de que ningún nodo es más influyente que otro, se pueden admitir nodos de configuración descentralizada (enlaces en gris), en los cuales se replica la *blockchain* en los nodos vinculados, pero estos no tienen relación alguna con el resto de la red distribuida. Elaboración propia.

Dentro de una red con *blockchain* los nodos pueden interactuar mediante la transferencia de *activos digitales*, para lo cual se requiere de la validación o aprobación de todos los participantes de la red informática (ver *Figura A.0.1*). Dicha validación se basa en algoritmos criptográficos para dar seguridad a las transacciones y constituye el *protocolo de consenso* para garantizar que, "ante los ojos de todos los miembros de la red" y sin necesidad de una autoridad central, se ha efectuado una transacción entre dos nodos [5].

Una vez que la transferencia se ha validado, los datos se almacenan en un bloque de información que constituye un "eslabón" de una *cadena* de transacciones que, debido a sus características especiales, es inalterable, permanente y rastreable [6][15] (ver *Figura A.0.2*). En efecto, cuando se validan las transferencias de *activos digitales* en la red, estas quedan grabadas en la *blockchain* sin que sea posible modificarlas. Luego, el hecho de que la *blockchain* sea inmutable permite construir un historial seguro de las transferencias realizadas que puede ser inspeccionado por todos los miembros que componen la red, facilitando la trazabilidad cronológica de los *activos* implicados [15][16].

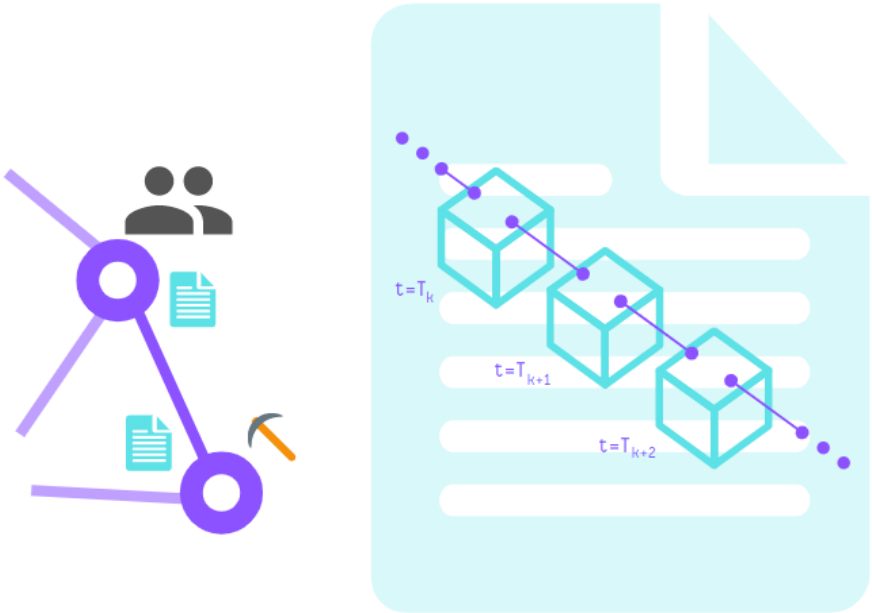


Figura A.0.2: Representación de la *cadena* que forman las transacciones validadas en una red con *blockchain*. Todos los nodos de la red tienen una copia actualizada de dicha base de datos, ya sean usuarios o *mineros*. Elaboración propia.

Por otra parte, un *smart contract* o *contrato inteligente* es un programa informático alojado en una red con *blockchain* que facilita el cumplimiento de un acuerdo entre dos partes a través de la ejecución automática de instrucciones lógicas previamente definidas en el sistema [6][15], permitiendo que los "activos digitales" se transfieran bajo ciertas condiciones propias de la red en la que operan.

Con todo lo anterior, en Chile, *Phineal* ha desarrollado *Blockchain GTIME*, que es una *blockchain federada* diseñada para la trazabilidad de inyecciones y retiros de energía eléctrica, sostenida por una red de medidores independientes llamados *phiNergy* y *phiNet* (nodos), los cuales, en intervalos de 15 minutos, levantan un vector de información que contiene la Geolocalización del medidor, el Tiempo en que se midió la energía, la ID del usuario, la MAC Address del dispositivo y la Energía que ha sido medida. Dicho vector de información se ha denominado "*GTIME*" en función de las iniciales de los atributos mencionados y equivale al *activo digital* que se transfiere en la red [26].

En base a *Blockchain GTIME* se desarrolló *Sello Sol*, que es una certificación de uso del recurso solar en tiempo real. Desde un principio, la idea ha sido que los consumidores finales tengan acceso a una plataforma web que garantice el uso de la energía solar en la elaboración de productos y/o servicios mediante la publicación de datos relativos a la producción energética de las plantas solares utilizadas para ello, independientemente del tipo de abastecimiento con el que cuente la empresa o institución (a saber, generación distribuida para autoconsumo o contrato de suministro con una generadora solar) [26].

En ese contexto, la Memoria de Título busca validar y fortalecer el sistema de trazabilidad de energía eléctrica de *Sello Sol* a través de la programación de un *smart contract* que reproduzca una *cadena de suministro* instantánea basada en *Blockchain GTIME*; siendo la planta PV Salvador el punto de generación y la oficina de *Phineal* el punto de retiro, incorporando el costo físico por transporte (pérdidas eléctricas). Dentro de las metodologías de asignación de pérdidas estudiadas formalmente en este documento se encuentran:

- Método de Bialek [1]
- Método de Kirschen, Allan & Strbac [2]
- Método de Prorrateo o Estampillado Postal [45]

En sintonía a la regulación chilena actual (artículo 115° de la Ley 20.936) [37], se decidió utilizar las metodologías que permiten desplazar las pérdidas hacia los clientes finales; tomando un modelo simplificado del Sistema Eléctrico Nacional (SEN), el que se extiende desde la barra Paposo hasta la barra Ancoa y en el cual la ciudad de Santiago se alimenta desde la barra Cerro Navia (ver *Figura A.0.3*).

En coherencia a lo anterior, y en perspectiva de la implementación de las metodologías sobre el prototipo tecnológico del sistema de trazabilidad con *blockchain*, se seleccionó el Método de Bialek con Flujos Gruesos para conseguir una *cadena de suministro* a nivel de transmisión nacional, mientras que se optó por emplear el Método de Estampillado Postal a nivel de transmisión zonal y distribución para extender la *cadena de suministro* hasta el cliente final.

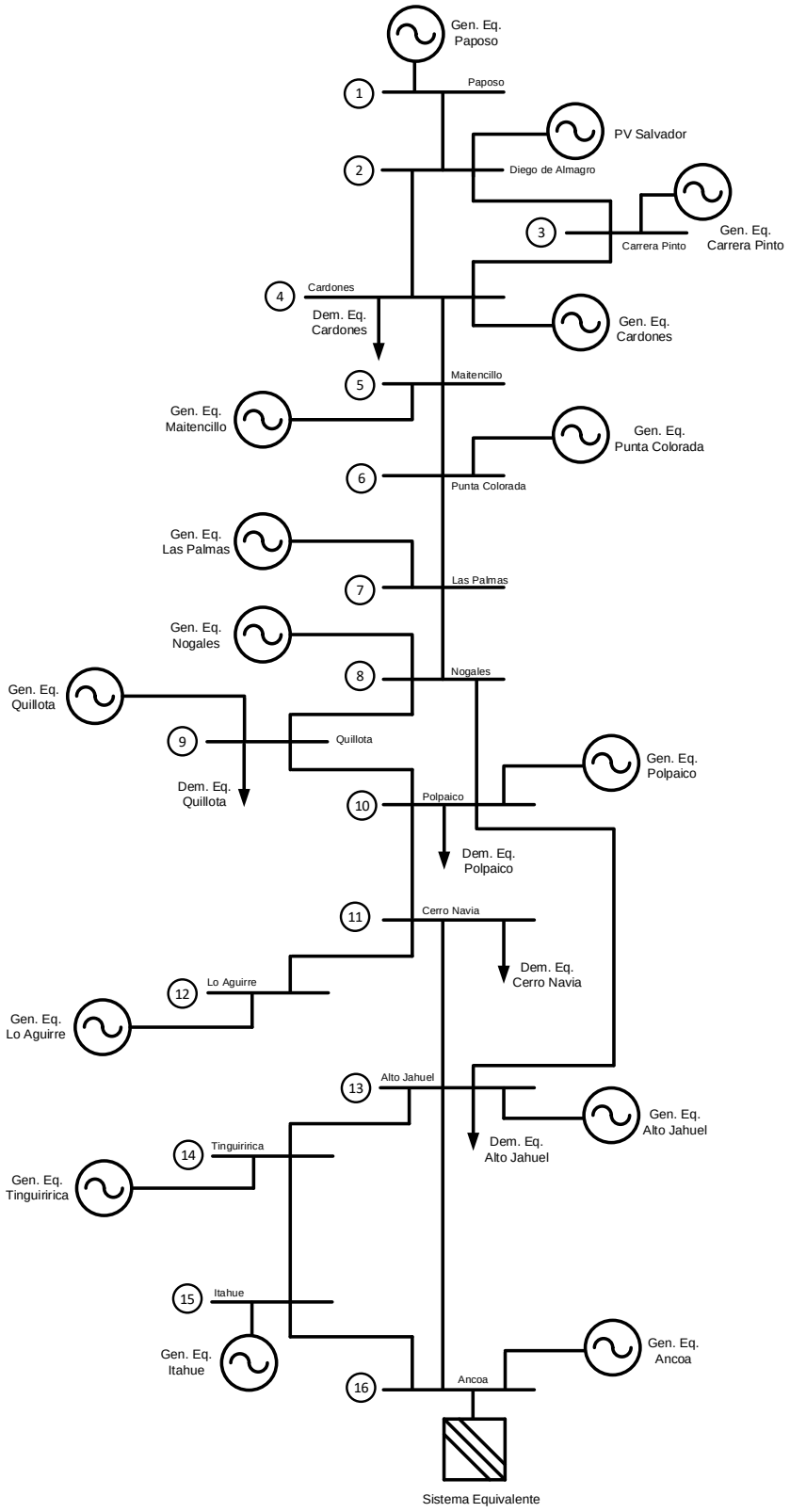


Figura A.0.3: Simplificación del SEN en coherencia a los criterios del problema de trazabilidad. En base a este esquema se aplicó el Método de Bialek con Flujos Gruesos, mientras que en la carga de Cerro Navia (barra 11) se aplicó el Método de Estampillado Postal. Elaboración propia.

Así, la combinación sinérgica de estos procedimientos en la propuesta metodológica permite conocer el nivel de participación de la planta PV Salvador en el abastecimiento y en las pérdidas asignadas a la demanda conectada en Cerro Navia y, en función de dichas pérdidas locales, permite prorratear el suministro y el costo físico de transporte en virtud de la proporción de consumo del cliente final en Santiago (ver *Figura A.0.4* y *Figura A.0.5*).

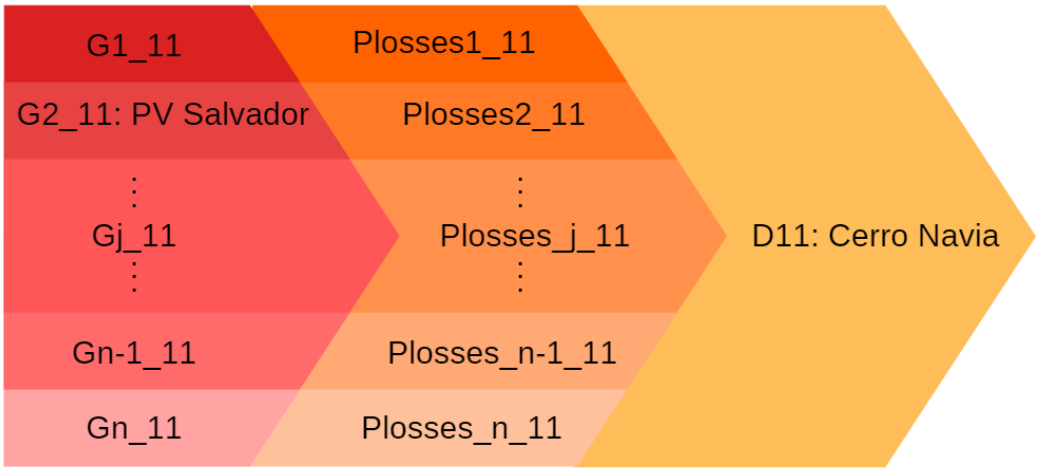


Figura A.0.4: Interpretación gráfica de la aplicación de Flujos Gruesos de J. Bialek en SEN reducido de la *Figura A.0.3*. El resultado permite apreciar quiénes son los generadores que alimentan a Cerro Navia y en qué cantidad, además de la magnitud de las pérdidas que produce cada uno de ellos a nivel de transmisión nacional. Elaboración propia.

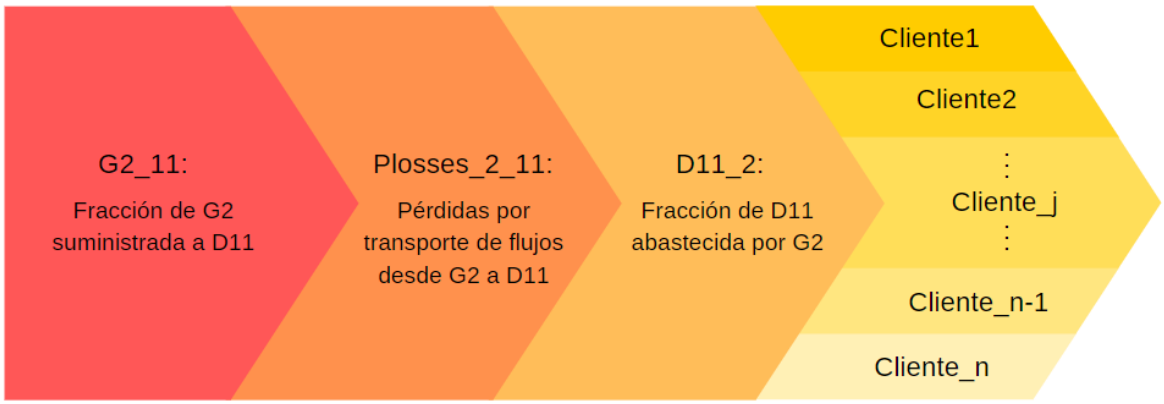


Figura A.0.5: Interpretación gráfica de la aplicación de Estampillado Postal en la barra Cerro Navia del SEN reducido de la *Figura A.0.3*. En función de la producción que PV Salvador destina a Cerro Navia y de las pérdidas incurridas a nivel de transmisión nacional por dicho concepto, se prorratea el abastecimiento y las pérdidas (desplazdas, ahora, a nivel de transmisión zonal y de distribución) según el consumo de los clientes finales. Elaboración propia.

Para implementar la propuesta metodológica sobre la *DApp* a fin de conseguir la *cadena de suministro* instantánea y experimental entre los agentes, se definió que los datos de inyección y retiro serían provistos por las mediciones de *Blockchain GTIME*; mientras que para el cálculo de las pérdidas instantáneas en Cerro Navia se decidió simular 14 estados horarios del modelo simplificado del SEN en los que se aplicó el Método de Bialek (estados formulados con datos empíricos levantados por los equipos de telemetría del Coordinador Eléctrico Nacional [38]). Luego, gracias a la aplicación del Método de Estampillado Postal presente en la *DApp*, fue posible obtener las proporciones instantáneas de generación, pérdidas y consumo de la *cadena de suministro* a nivel del cliente final.

En lo que respecta a la *DApp*, esta se apoya en el uso de dos archivos fundamentales: un *smart contract* titulado "`EnergyTraceability.sol`" que hace de gestor de los datos de energía; y de un archivo *JavaScript* titulado "`app.js`" que se ocupa de formular la *cadena de suministro* y de mostrarla a través del navegador web (en la dirección de "loopback" o *localhost*, para efectos de prototipo).

El *smart contract* se despliega sobre una red de prueba de *Ethereum* provista por el "framework" de desarrollo Truffle (red "TestRPC"). Para importar los datos desde fuentes externas a red de *Ethereum*, el contrato utiliza una herramienta de consulta web llamada "*Oraclize*". Cuando el *smart contract* invoca a *Oraclize*, esta recurre simultáneamente a distintas plataformas de extracción de datos web, permitiendo el rescate de la información de forma descentralizada, esto es, evitando que *Oraclize* se transforme en un intermediario en el que haya que depositar confianza [42]. Se debe tener en cuenta que cada solicitud y respuesta de datos implica una transacción descentralizada en la *blockchain pública* de *Ethereum* y, como tal, esta debe validarse mediante un *protocolo de consenso* definido en la red.

En virtud de la implementación de la propuesta metodológica en la *DApp* (explicada anteriormente), el *smart contract* recurre a la función "`update()`" para solicitar los datos a través de la web, que son:

- Inyección de la planta y retiro del cliente final; medidos por los nodos *phiNet* y *phiNergy* de *Phineal*. Datos extraídos desde el *Server* de *Blockchain GTIME* (unidad de almacenamiento).

- Factor de proporción de generación horaria de PV Salvador, *demanda gruesa* horaria y *demanda real* horaria; calculados a partir de la aplicación del Método de Bialek sobre el modelo simplificado del SEN en los 14 estados horarios ya descritos. Datos extraídos desde un servidor ficticio del Coordinador (dispuesto para este prototipo).

Una vez que dispone de los datos solicitados, el *smart contract* discrimina la fuente de la respuesta con la función "`callback()`". De este modo, la *DApp* "entiende" si los datos provienen desde *Blockchain GTIME* (y, por tanto, hay que tratarlos como inyección y retiro) o si vienen desde el servidor ficticio del Coordinador (en cuyo caso hay que trabajarlos para calcular las pérdidas eléctricas de estado).

En cualquiera de los dos casos, el archivo *JavaScript* es quien se hace cargo de la siguiente etapa, recibiendo los datos "clasificados" desde el *smart contract*, aplicando las funciones "`LogEnergyUpdate.watch()`" y "`LogLossesUpdate.watch()`" y desglosando la información para asignar las cuantías recibidas sobre variables eléctricas coherentes. Luego, a través de la función "`showBalance()`", el archivo "`app.js`" efectúa los cálculos para formular la *cadena de suministro* a nivel nacional y, una vez resuelta la proporción de consumo del cliente final, expresa la *cadena de suministro* a nivel zonal y de distribución.

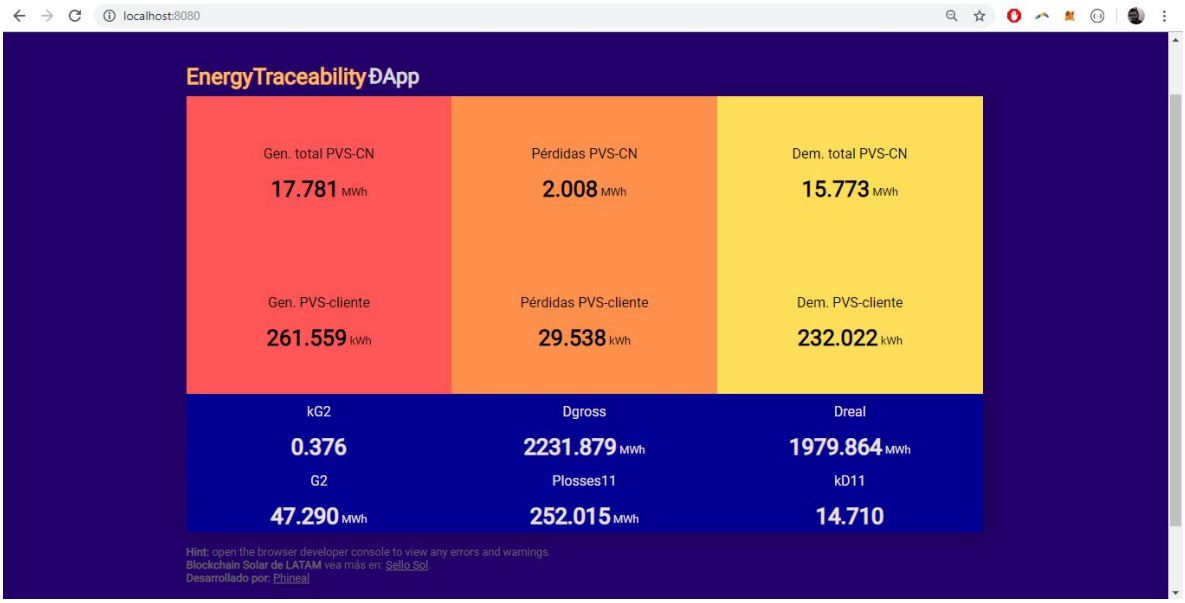


Figura A.0.6: Despliegue de la *DApp* en la dirección de "loopback" de un navegador web. Se aprecia la *cadena de suministro* instantánea a nivel de transmisión nacional (arriba) y a nivel del cliente final (abajo). Los parámetros incluídos en el cuadro azul corresponden a los resultados intermedios de la propuesta metodológica y sólo se incluyen para fines demostrativos. Elaboración propia.

Finalmente, con ayuda de la función "`showBalance()`" y gracias a un *Webpack* disponible en la *DApp* (archivos extraídos desde la web colaborativa [53] y adaptados a los objetivos de este proyecto), la trazabilidad de flujos termina exhibiéndose en la dirección *IP* de "loopback" o *localhost*; donde se puede apreciar, a modo de prototipo, el estado de la inyección, de las pérdidas y del retiro entre los agentes del problema.

De este modo es posible conseguir, para un determinado instante de tiempo y en rangos de 20 minutos, una *cadena de suministro* instantánea de electricidad que cuenta con dos niveles de validación en sus datos de origen (*Blockchain GTIME* y *Ethereum*), para su uso en certificados de producción, de abastecimiento y de *cadena de suministro* de "energía verde".

8.1 Anexos Capítulo 1

...[ ]...

8.2 Anexos Capítulo 2

8.2.1 Minería de datos, protocolo de consenso y bifurcaciones en la red

En los ítems precedentes se hizo hincapié en que una de las características más distintivas de la tecnología *blockchain* es el "consenso" que, como se ha dicho, está vinculado a los conceptos de *minería* y *hash* de los bloques. Pero, ¿Qué es la *minería* en *blockchain*?, ¿cómo funciona el mecanismo que dota de seguridad a la red distribuida?

La *minería* de datos, según se expone en "www.bitcoin.org", constituye un proceso de cálculos matemáticos a través de computadores para confirmar las transacciones que se incorporarán a la *blockchain*. De lo planteado por T. Fernández y P. Fraga [16], en tanto, se desprende que la *minería* de datos permite, dentro de los límites del *protocolo de consenso* convenido, cumplir el orden cronológico de los bloques, aumentar la seguridad de la red y conseguir un acuerdo global respecto del estado del sistema, sin que sea necesaria la intervención de una entidad validadora externa. Este proceso puede sostenerse mediante recompensas para los *mineros* (en caso de que cualquier usuario pueda modificar la base de datos como en una *blockchain pública*; a saber: *Bitcoin*, *Ethereum* o *Litecoin* [17]), o bien, a través de contratos previos con la comunidad validadora (en caso de que los usuarios estén perfectamente identificados y autorizados para editar la base de datos como en una *blockchain privada*; por ejemplo: *Hyperledger-Fabric*, *Corda* o *Quorum* [17]).

Según estipula S. Nakamoto en su artículo [5], y tal como se mencionó en el ítem 2.1.2, la *blockchain* de *Bitcoin* basa su consenso en la *prueba de trabajo*, que básicamente consiste en la búsqueda de un número llamado *nonce* que, una vez ingresado en el argumento de la función criptográfica SHA-256, permite obtener el *hash* del "bloque postulante" que cumple con la condición de que su código alfanumérico comience con un número definido de bits con cuantía "0".

Dicho requerimiento es propio de las *blockchains* que operan con *prueba de trabajo* y está vinculado a otro concepto llamado *dificultad de minado*, con el cual el sistema pretende "compensar el aumento en la velocidad del hardware y el interés variable de los nodos activos a lo largo del tiempo, de modo que, si los bloques se generan muy deprisa, la *dificultad* aumenta" [5].

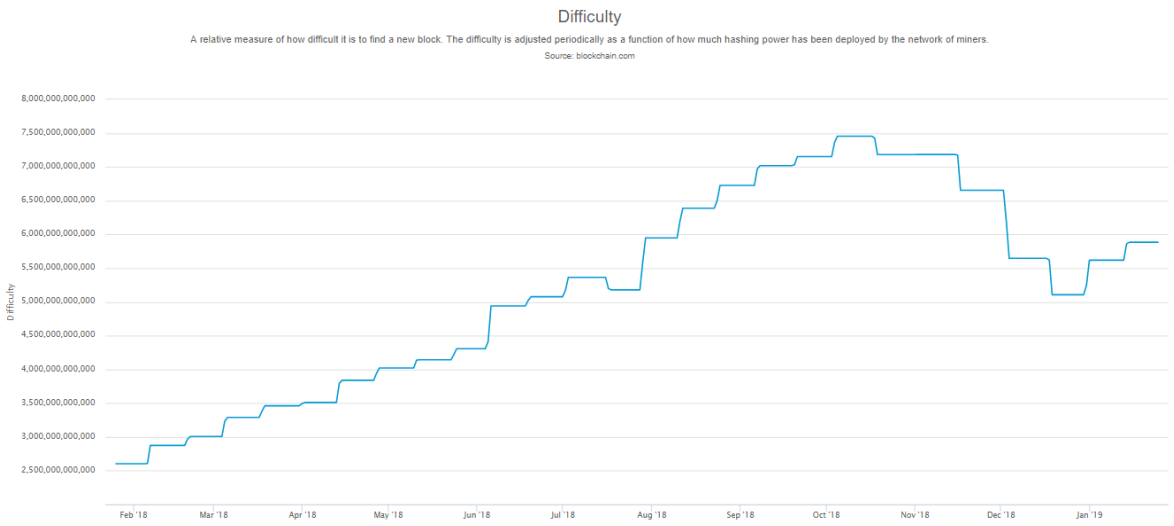


Figura A.2.1: Dificultad de minado en *blockchain pública* de *Bitcoin* [46].

En términos concretos, acertar la respuesta para un *hash* que comienza con tres "0" es más simple que hacerlo para uno que comienza con treinta "0". Lo anterior implica que *minar* con un computador ordinario es razonable cuando la *dificultad* del sistema es baja, pero para procesos como la construcción de la cadena de *Bitcoin*, por ejemplo, conviene *minar* con equipos más potentes o en *pools* (conjunto de máquinas de *minado* trabajando bajo el mismo nodo), debido a que la exigencia computacional es mucho mayor. La *Figura A.2.1* es información gráfica publicada en la web "blockchain.info" y exhibe la evolución de la *dificultad* de la *blockchain* de *Bitcoin* durante el último año (febrero 2018 – febrero 2019).

En ese sentido, uno de los factores determinantes es el hardware que se utiliza para llevar a cabo el *minado* de bloques. Entre las cualidades más relevantes destacan: el precio de los equipos (que va desde cientos a miles de dólares dependiendo de las exigencias de la *blockchain* en la que se desee trabajar, ya que podrá variar desde equipos con potentes procesadores CPU, pasando por los que sólo usan tarjetas gráficas de elevada potencia GPU, hasta llegar a los circuitos integrados de aplicaciones específicas ASIC), la tasa de *hash* o *hash rate* (que, en coherencia a lo anterior, podrá ser de unos cuantos "Kilo-*hash*/segundo" con un buen CPU, hasta varios "Tera-*hash*/segundo" para aquellas máquinas con ASIC), el consumo eléctrico de los equipos y el algoritmo a minar (entre los que se encuentran el SHA256, Scrypt, Ethash, X11, Equihash, Cryptonight, entre otros).

Al respecto, la gráfica de la *Figura A.2.2* corresponde a la evolución del *hash rate* de la *cadena de bloques* de *Bitcoin* desde febrero de 2018 hasta febrero de 2019 y también ha sido extraída desde la web "blockchain.info". Las imágenes que siguen, en tanto, pretenden mostrar el aspecto de un equipo de

minado típico entre los nodos validadores de *Bitcoin* (GPU en *Figura A.2.3*) y el de una "granja" de mineros operando en Rusia (ver *Figura A.2.4*).

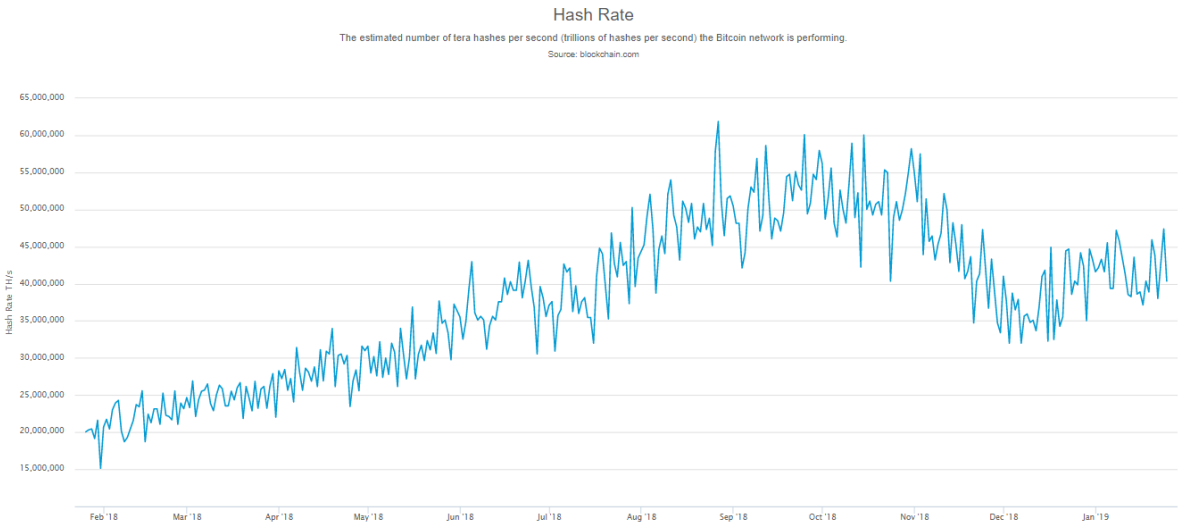


Figura A.2.2: Tasa de cálculo de *hash* por segundo (del orden de TH/s) en *blockchain pública* de *Bitcoin* [46].

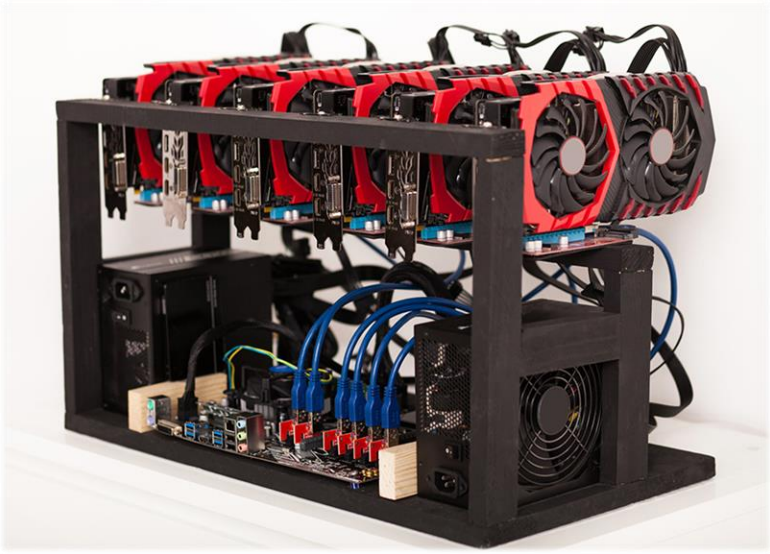


Figura A.2.3: Nodo minero tipo GPU, modelo "AMD Rig 6 Vga RX470". Puede minar 14 algoritmos distintos a una tasa máxima de 156MH/s, consumiendo 720W de potencia [48].

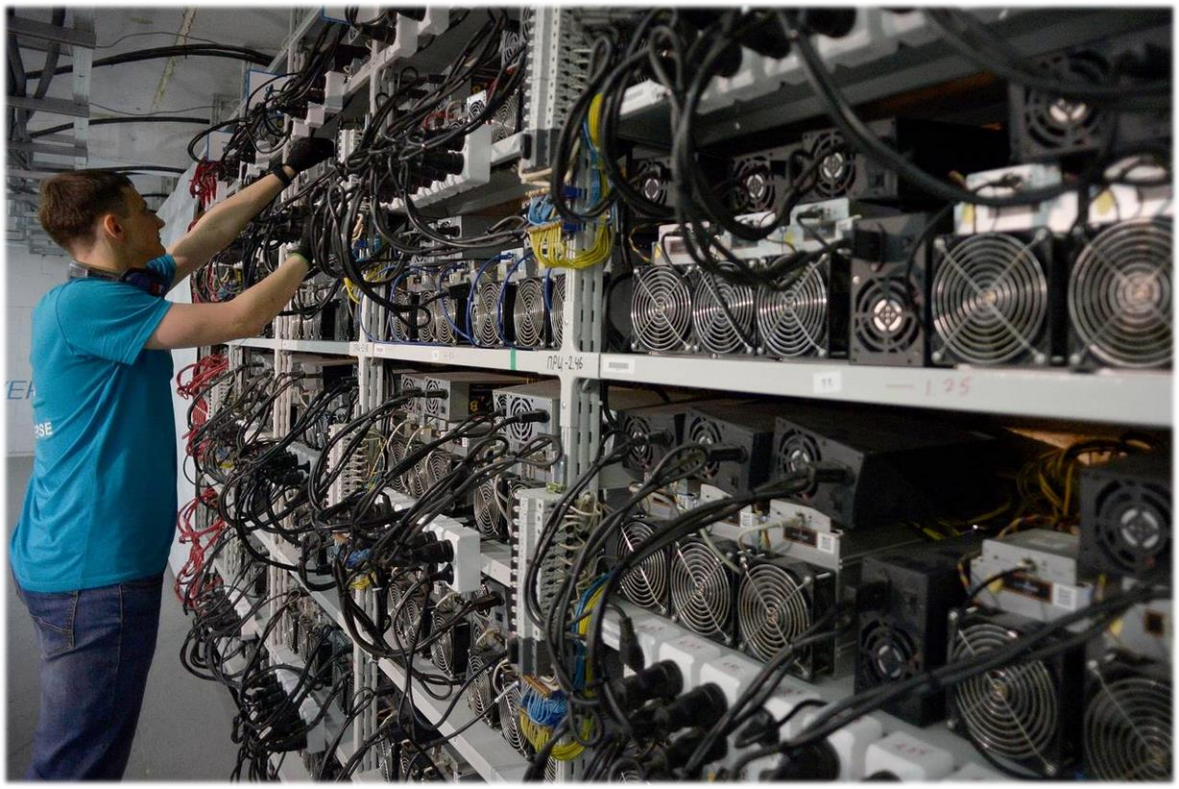


Figura A.2.4: Granja de minado "CryptoUniverse" en Rusia, para sostener la *blockchain* de *Bitcoin* y *Litecoin* [49].

Para ilustrar el proceso de *minado* de bloques, se ha recurrido a la web "Blockchain Demo" [47] que presenta un modo amigable de entender cómo se va creando una *cadena de bloques*. A continuación se exhibe el proceso mediante un ejemplo didáctico:

- La cadena comienza con un elemento como el de la *Figura A.2.5*, que independientemente de que haga referencia al *bloque génesis*, expone la composición típica de un bloque de *blockchain*, a saber: *índice*, *marca de tiempo*, *hash*, *hash anterior*, *datos transados* y *nonce*.

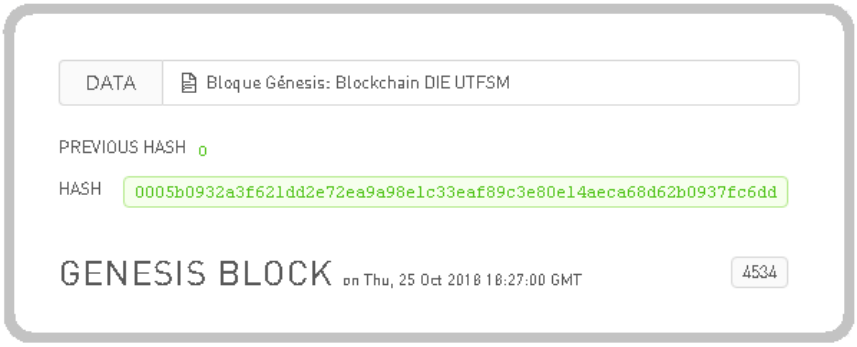


Figura A.2.5: Bloque Génesis en la aplicación didáctica de la web "Blockchain Demo" [47].

El *índice* hace referencia la posición del bloque en la cadena, en este caso, para el *bloque génesis* el *índice* es igual a "0" (no mostrado en la imagen anterior), mientras que para el próximo bloque valdrá "1". La *marca de tiempo*, por su parte, viene expresada con la frase "*on Thu, 25 Oct 2018 18:27:00 GMT*" y será codificada para su posterior utilización. Si se pone atención a los *datos transados*, se observará que en este caso son letras y están asociadas a la frase: <<*Bloque Génesis: Blockchain DIE UTFSM*>>. Adicionalmente, el número *nonce* (ubicado en la parte inferior derecha de la imagen) es el valor que un nodo postula, mediante un proceso iterativo, para dar con el *hash* válido correspondiente. Como se ha dicho antes, el proceso de búsqueda de dicho *nonce* se llama *minería* y el nodo que propone dichos valores se llama *minero*. La *Figura 2.6* también muestra el *hash*, que como se ha enfatizado a lo largo de este capítulo, es un código alfanumérico de longitud fija que hace las veces de "huella digital" de cada bloque y nace a partir de una función criptográfica f_{cript} previamente definida en el sistema. Dicha función, que depende de la *blockchain* en la cual se trabaje (SHA-256 en *Bitcoin*, por ejemplo [5]), utiliza el contenido del bloque como argumento para generar el número *hash*. En este caso, dado que no hay bloques antes del *bloque génesis*, el *hash anterior* no existe y con ello se puede escribir (ver *Figura A.2.5*):

$$f_{cript}(\text{índice}, hash\ anterior, \text{cód.}\ marca\ de\ tiempo, datos, nonce) = hash_{génesis}$$

$$\Rightarrow f_{cript}(0{,}0{,}251028000000, \ll \textit{Bloque Génesis: Blockchain DIE UTFSM} \gg, 4534) = hash_{génesis}$$

$$\therefore hash_{génesis} = 0005b0932a3f621dd2e72ea9a98e1c33eaf89c3e80e14aeca68d62b0937fc6dd$$

Esto implica que existe un único *hash* para el contenido particular de un bloque, de modo que, si el argumento de la función cambia, el *hash* experimenta cambios importantes en su cifrado (de manera consistente a lo expuesto en el ítem 2.1.2). Adicionalmente, si bien es posible conocer el *hash* de un grupo de datos, no es factible saber el contenido asociado a un *hash* específico. También, en la imagen anterior se nota que la *dificultad* es de tres dígitos "0", dado que el *hash génesis* comienza con tal cantidad de cifras.

- Para ejemplificar lo que se ha dicho sobre el *hash*, la web "blockchaindemo.io" [47] propone hacer cambios en el contenido del bloque para ver los efectos sobre el *hash*. En este caso, se decidió cambiar dos caracteres de los *datos transados*, quedando: <<*Bloque Génesis: Blockchain DIE USM*>>. El resultado es como se muestra en la *Figura A.2.6.*

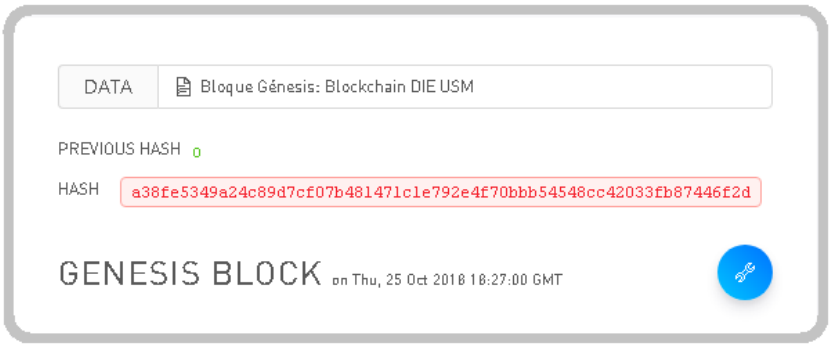


Figura A.2.6: Bloque Génesis con datos manipulados. Aplicación didáctica de la web [47].

Se observa de inmediato que el *hash* del *bloque génesis* ha cambiado por completo. Esto dejaría totalmente inválido el conjunto de bloques subsiguientes. Dado que la página tiene fines didácticos, se tiñe en rojo el *hash* para mostrar que este no es válido y se insta al usuario a "buscar" el nuevo *nonce* para *minar* correctamente el bloque (equivale a "simular" que se hace un proceso iterativo hasta encontrar el *hash* correcto). Con lo anterior, si se valida este nuevo *bloque génesis* y luego se añaden dos bloques más, se tiene el esquema mostrado en la *Figura A.2.7*, de donde se desprende que la *blockchain* se articula perfectamente gracias a los conceptos revisados en el ítem 2.1.2, garantizando la integridad de los datos. Aquí se debe entender que para conseguir que los dos últimos bloques se agregaran a la cadena, fue necesaria la respectiva validación de la comunidad de observadores, quienes han comprobado, a través de los *hashes* despejados, que los números *nonce* propuestos permiten "sellar" correctamente los bloques recién adheridos, es decir, se ha llegado a un consenso. Luego, todos los *mineros* han de actualizar su copia de la *blockchain* para replicar dichos valores en su registro y seguir con el proceso de concatenación. Por último, se propone adherir un nuevo fichero con objeto de crear una cadena de tres bloques. A continuación, se cambia el contenido de uno de ellos para observar cuál es el impacto en el sistema (como si se tratara de una operación intervenida, "hackeada" o fraudulenta). El resultado se expone en la *Figura A.2.8*.

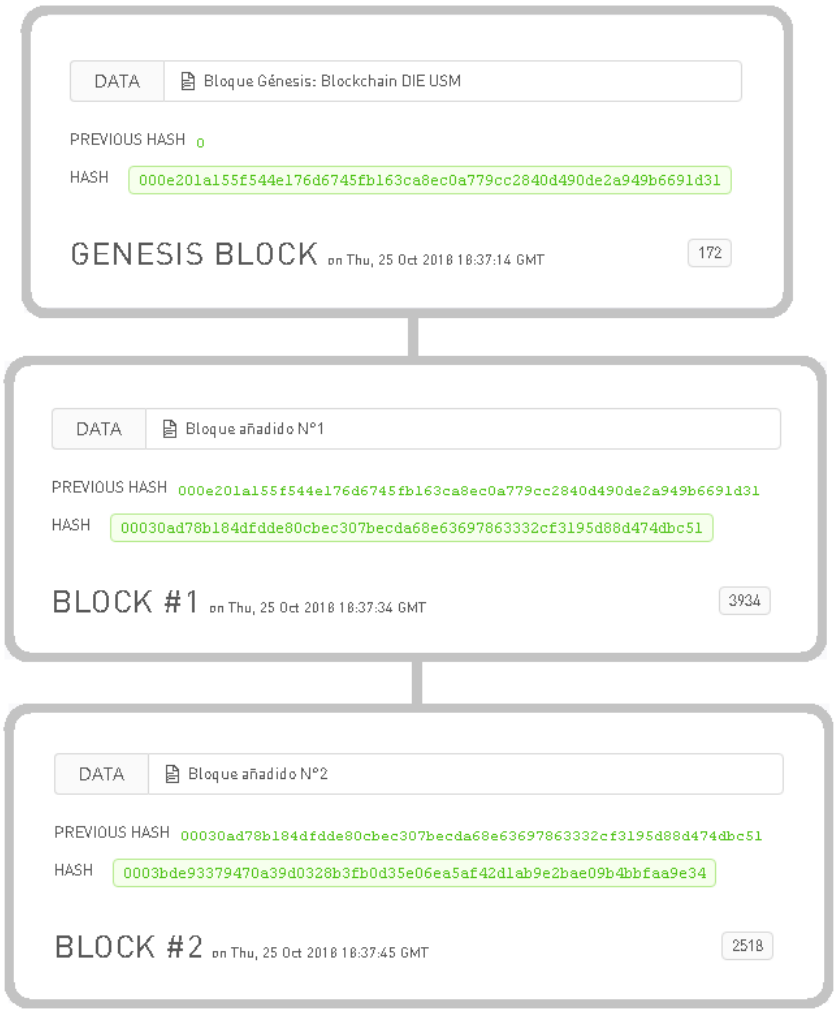


Figura A.2.7: Cadena de bloques enlazada correctamente. Aplicación didáctica de "Blockchain Demo" [47].

Como puede notarse, las consecuencias de cambiar uno de los bloques enlazados repercute en que se invalidan todas las transacciones posteriores (el contenido del *Bloque #2* ha sido cambiado a: *<<Bloque añadido N°2 – HACKEADO>>*). En tal caso, los nodos rechazan los bloques erróneos y la única salida es *minar* de nuevo las transacciones para que sean aceptadas por la comunidad de observadores. De este modo, el *hash* intervenido destruye toda posibilidad de enlace entre los bloques subsiguientes, evidenciando que no existe consenso respecto de esta versión alterada de la cadena.

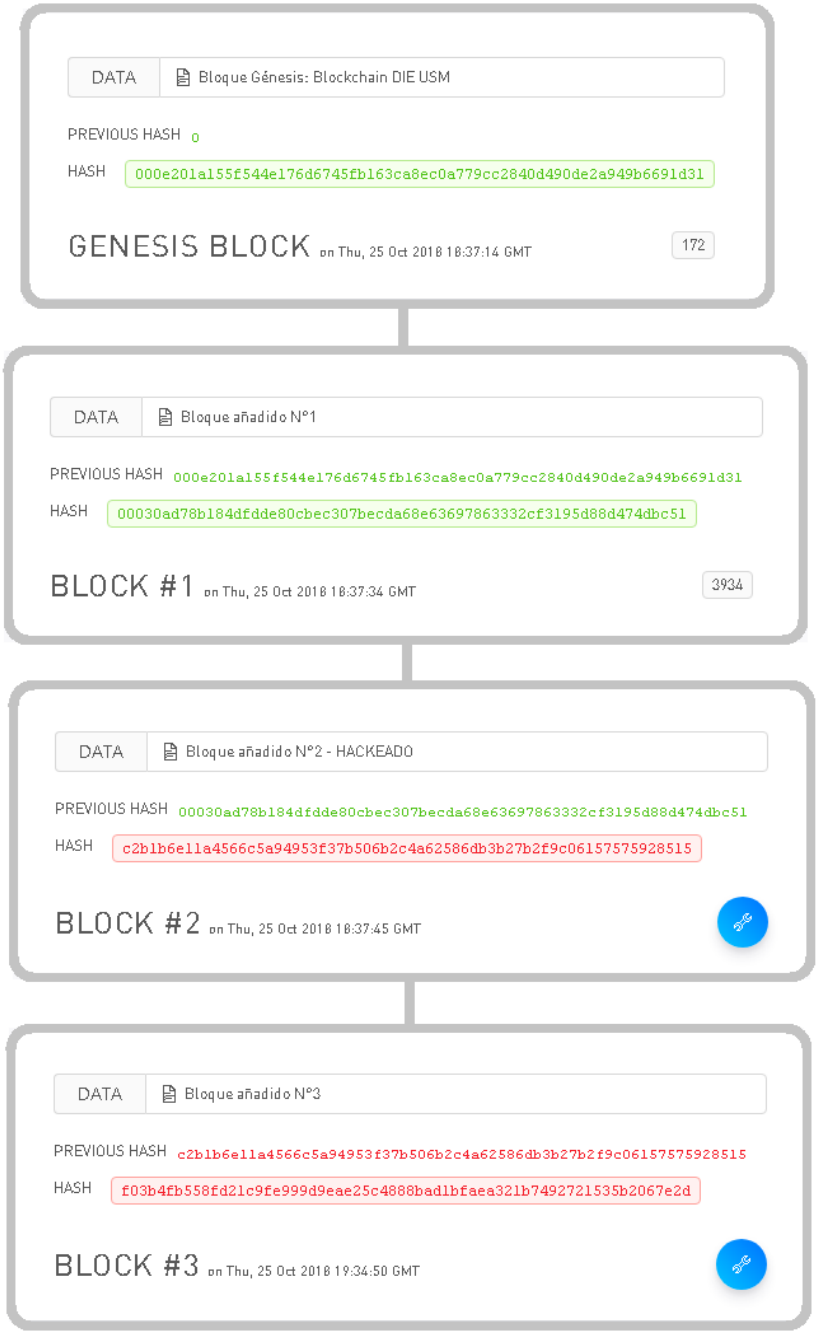


Figura A.2.8: Cadena de bloques interrumpida por la alteración de los datos transados [47].

En suma, gracias a la *minería* de datos se garantiza la inmutabilidad de la tecnología *blockchain*, ya que un bloque no puede ser editado a menos que se realice todo el trabajo de enlace y verificación nuevamente. Ahora bien, con los conceptos previos resueltos es válido preguntarse: ¿cómo se logra el

consenso en una red distribuida de usuarios entre los cuales no existe confianza?, ¿cómo resuelve *blockchain* el problema de desconfianza en la validación de las transacciones?

Para contextualizar, quizás valga la pena recordar que una red informática tradicional está compuesta por un servidor central y por usuarios que pueden leer los datos allí almacenados. En este caso, si bien el servidor central es el único que puede modificar su propio registro (otorgando seguridad y confianza sobre la información guardada), este esquema es altamente vulnerable pues una falla en el servidor bloquea por completo el funcionamiento de la red.

Al respecto, una solución es contar con múltiples servidores que tengan una copia exacta de la base de datos; de modo que, si uno de los servidores falla, la información sigue disponible en alguna de las réplicas. No obstante, como la transmisión de datos no es inmediata, es posible que en ciertas ocasiones exista más de una versión de la base de datos. Esto último obliga a definir ciertas reglas para diferenciar cuál es la "información oficial", o dicho de otro modo, reglas que permitan establecer un *protocolo de consenso* ante discrepancias repentinas en la base de datos.

En consecuencia, el problema de consenso en redes distribuidas en las que no existe confianza entre sus participantes hace referencia a la dificultad de conseguir la "versión aceptada" de la información frente a escenarios en los que cualquier usuario puede conectarse y hacer modificaciones sobre la base de datos, afectando potencialmente la seguridad del sistema.

Concretamente, para mantener el *protocolo de consenso* en *blockchain*, la comunidad de nodos interactúa constantemente entre sí para determinar cuál es la cadena más actualizada. El diagrama de bloques "tipo" de este proceso de interacción entre nodos puede hallarse en la web [47] y es como se muestra en la *Figura A.2.9*.

Nótese que el esquema de la *Figura A.2.9* sugiere que, aún cuando los nodos de una red con *blockchain* interactúan de manera *peer to peer* para actualizar la cadena y validar sus transacciones, pueden nacer inconvenientes.

En ese sentido, una de las situaciones conflictivas surge cuando, por casualidad, dos *mineros* postulan bloques diferentes de manera simultánea. Aquí, el consenso en la red se "quiebra" y la *cadena* experimenta una falla *bizantina* o *bifurcación* (*fork*); lo que implica que existan dos versiones diferentes de la información concatenada (ver *Figura A.2.10*). Ante esto, cada *minero* deberá decidir en cuál de las ramificaciones seguirá adicionando bloques, al menos hasta que el *protocolo de consenso* diga lo contrario.

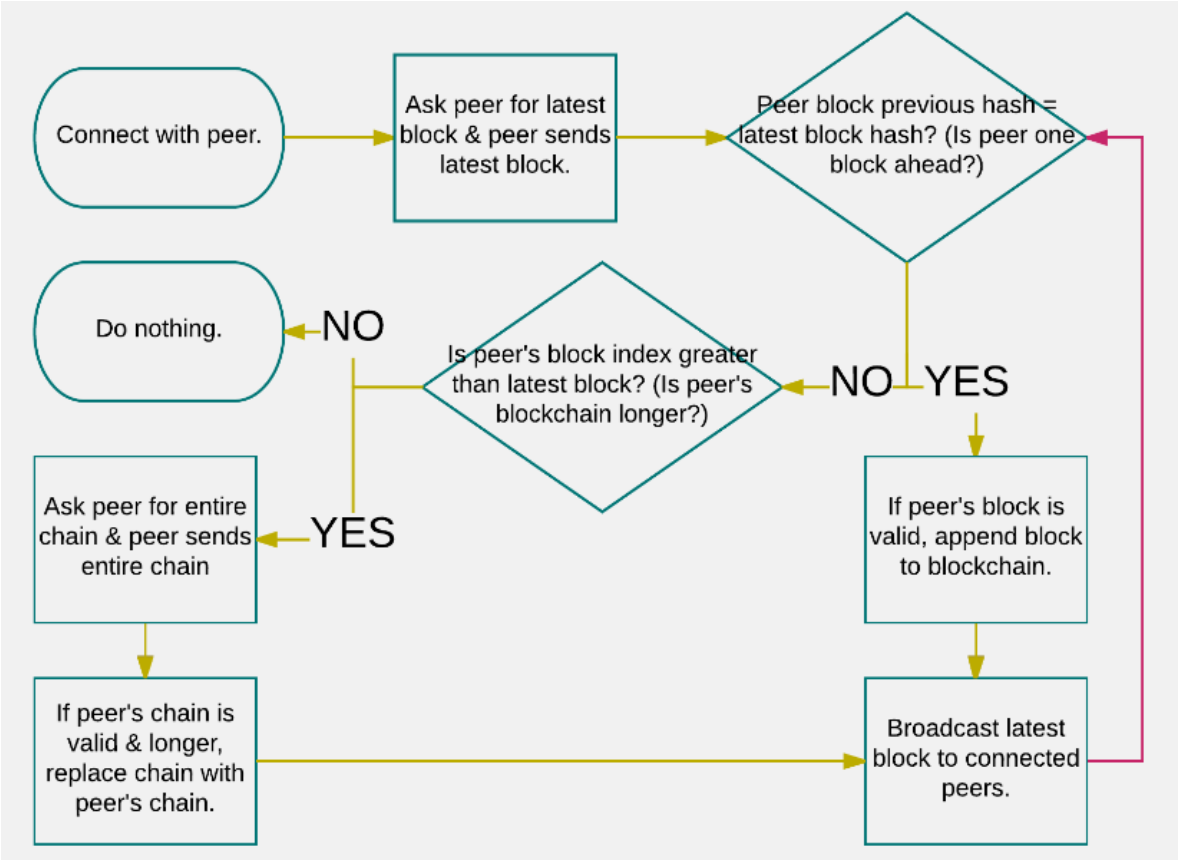


Figura A.2.9: Diagrama de bloques que describe la actualización de la *blockchain* en cada nodo de la red en [47].

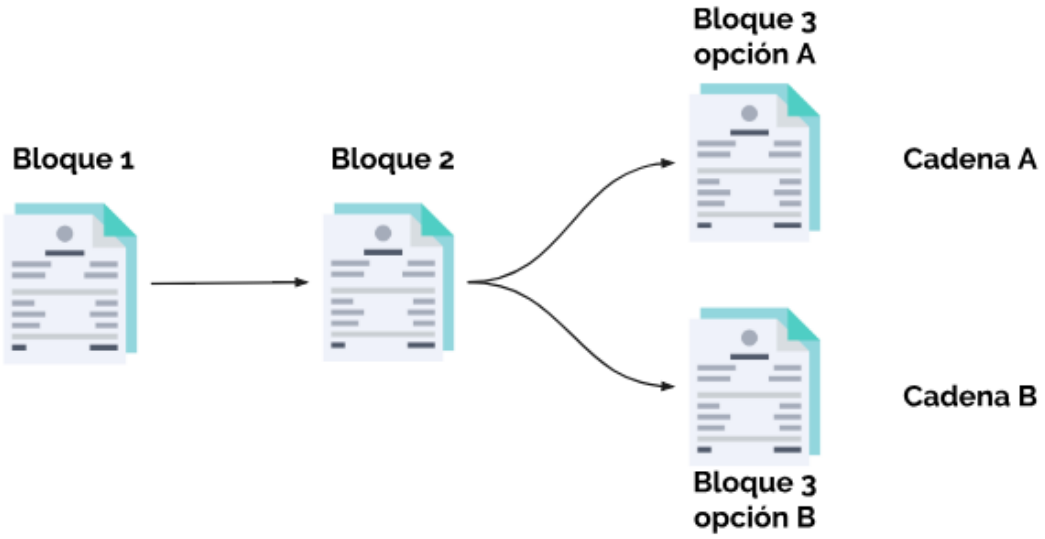


Figura A.2.10: Esquema de representación de la ocurrencia de *fork* en una *cadena de bloques* [12].

Como se ha dicho antes, *Bitcoin*, por ejemplo, basa su consenso en la *prueba de trabajo* o *PoW*; por lo que ante una *bifurcación*, el sistema optará por la rama más larga, o equivalentemente, en la cual se ha invertido el mayor esfuerzo computacional de parte de los nodos. De este modo, explica Nakamoto, "si la mayoría de la potencia CPU está controlada por nodos honestos, la cadena honesta crecerá más rápido y dejará atrás cualquier cadena que compita" [5].

Se habla de usuarios "honestos" porque una *bifurcación* puede ser una oportunidad para que se produzca un fraude, no obstante, para que este resulte exitoso, el "hacker" tendría que rehacer la *prueba de trabajo* del bloque en el que se produce el *fork* para luego construir la cadena hasta alcanzar el trabajo de los "nodos honestos", lo que es computacionalmente muy costoso. Así, la red es más segura cuanto mayor es la cantidad de *mineros* que mantienen la red [5], puesto que los algoritmos de consenso de *PoW* se basan en el hecho de que si un nodo realiza mucho trabajo para la red, es menos probable que este ataque la *blockchain* [16]. Esto último hace que *blockchain* también sea altamente resistente a ataques tipo "Sybil", en los que se irrumpe la red con una gran cantidad de transacciones erróneas o fraudulentas como consecuencia de que buena parte de los nodos validadores son propiedad de un único usuario [16].

Respecto a los *protocolos de consenso*, de acuerdo con lo que plantean T. Dinh, R. Liu, *et al.* [15], estos pueden clasificarse dentro de un espectro bien definido. En un extremo están los *protocolos basados en cómputo*, que usan la prueba de cómputo para seleccionar al azar un nodo que decide autónomamente su operación y en donde, ante una *bifurcación*, los *mineros* que trabajan sobre alguna de las ramas en realidad están emitiendo una especie de "voto" que tiene un peso proporcional a su poder de procesamiento (como la *PoW* de *Bitcoin*, por ejemplo). Esto último es bastante común en *blockchains públicas*, en donde el software es de código abierto y los nodos tienen acceso libre a la red.

Aquí es bueno agregar que, tal como comentan los autores de [6], se pueden usar otras funciones para obtener el *hash* además de la SHA-256 utilizada por *Bitcoin* en su *PoW*, como la función Blake-256 o la Scrypt usada en la criptomoneda *Litecoin*. También se habla de la *prueba de participación* (*proof of stake* o *PoS*) utilizada por *Ethereum*, que es una alternativa a la *prueba de trabajo* y que se basa en dar prioridad a aquellos nodos que poseen mayor cantidad de criptomonedas *ethers* (ETH), requiriendo menos cálculos de CPU para efectuar la *minería*.

Siguiendo con el espectro propuesto en [15], en el otro extremo del intervalo se encuentran los *protocolos basados en comunicación*, en donde los usuarios tienen "votos" iguales y pasan por múltiples rondas de comunicación para llegar a un consenso (como en el *protocolo Practical Byzantine Fault Tolerance* o *PBFT*, entre otros). Estos protocolos se usan en configuraciones donde se necesitan nodos autenticados o de identificación plena, como en *blockchains privadas*, en donde no se necesitan mecanismos de consenso costosos como la *prueba de trabajo*, pues el riesgo de un ataque de tipo "Sybil" no tiene cabida en estas redes [6].

Por su parte, mediante la cita de varios autores, la publicación [15] afirma que dentro del espectro de *protocolos de consenso* también viven *protocolos híbridos* que apuntan a mejorar el rendimiento de *PoW* y *PBFT*. Algunos como *Proof-of-Elapsed-Time* (*PoET*), *Elastico* y *Algorand* se utilizan en la mejora de *blockchains públicas* para abordar la ineficiencia de *PoW* mediante estrategias de software confiable y muestreo aleatorio de nodos, respectivamente.

Otros *protocolos híbridos*, en cambio, persiguen eliminar los incentivos económicos que sostienen la operación del *minado* de datos, brindando una gama más amplia de *protocolos de consenso* para elegir [6]. Estas iniciativas se enfocan en el mejoramiento de *blockchains privadas* que ocupan *PBFT*, a través de la ejecución de consensos en redes más pequeñas llamadas *redes federadas* o *dedicadas* que, según otros autores como [17], representan una nueva categoría de *blockchain*. Ejemplos de ello son: *Proof-of-Authority* (*PoA*), *Stellar* y *Ripple* [15].

8.2.2 Puntos fuertes y débiles de blockchain

8.2.2.1 Ventajas

Como se argumenta en las publicaciones [6] y [7] (a través de la cita de varios autores), independientemente del rango de acción que tiene el usuario o de la naturaleza de administración y disponibilidad de datos que tiene la red, un sistema basado en *blockchain* ofrece, como mínimo, los siguientes beneficios:

- Un sistema *peer to peer*, verdaderamente distribuido, descentralizado y tolerante a las fallas de los nodos. Se delega la objetividad a la lógica matemática implícita en la ejecución de los códigos, no a organizaciones humanas.

- Una red capaz de identificar *bifurcaciones* y conflictos de consenso, que luego resuelve mediante la convergencia a una única versión de la información aceptada.

- Transparencia, verificación y auditoría de la actividad de la red. El intercambio de un *activo digital*, así como la aplicación automática de cláusulas en un *contrato inteligente*, supone que cualquier usuario de la red puede verificar y consultar el histórico de las transacciones registradas. Esto constituye una prueba auditable y pública de los movimientos en la red, eliminando la posibilidad de disputa entre las partes.

- Un sistema que favorece la interacción predecible y segura de usuarios que no confían entre sí. Una vez aplicado el *protocolo de consenso* sobre un bloque, la inmutabilidad de los datos es total, reduciéndose las posibilidades de fraude y malversación.

- Reducción de los costos de operación en los procesos; eliminando tareas de registro, control y repartición de datos, lo que conlleva directamente a la optimización de personal.

8.2.2.2 Desventajas

De acuerdo con los argumentos de J. Todó [7] y de V. Gatteschi, F. Lamberti, *et al.* [18], las virtudes de *blockchain* no tienen el mismo potencial en todos los sectores, desafiando a los desarrolladores de aplicaciones a mejorar diversos inconvenientes técnicos y económicos en esta tecnología. En consecuencia, los eventuales proyectos que adopten la *blockchain* en sus procesos y/o soluciones, deberán evaluar cuidadosamente las siguientes dificultades vigentes:

- El consumo energético es quizás el punto más débil de la tecnología *blockchain*. Para mantener la base de datos de una *blockchain pública* como *Bitcoin*, por ejemplo, el *minado* de bloques vía *PoW* tiene un elevado costo energético, llegando a 61,4 [TWh] en abril de 2018 según la revista Forbes (equivalente al 1,5% del consumo anual de EE.UU.) [19]. Ciertamente *Bitcoin* representa la aplicación más ineficiente de las *blockchains* hoy en día. De acuerdo con un artículo publicado en "www.digiconomist.net", la evolución del índice de consumo energético y las estadísticas clave para dimensionar el impacto ambiental de *Bitcoin* durante el último año (abril 2018 – abril 2019), son como se exhiben en las imágenes adjuntas.

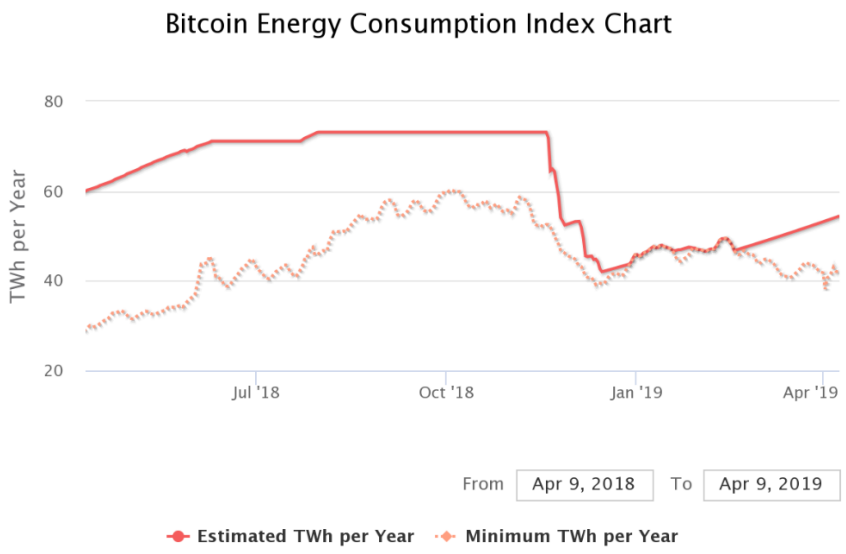


Figura A.2.11: Consumo energético del total de mineros de la red *pública* de *Bitcoin* según Digieconomist [50].

Descripción	Valor
Consumo de electricidad anual estimado actual de Bitcoin * (TWh)	54.44
El consumo de electricidad mínimo anual actual de Bitcoin ** (TWh)	41.86
Ingresos mineros globales anualizados	$ 3,846,431,884
Costos mineros globales anualizados estimados	$ 2,721,866,758
Porcentaje de costo actual	70.76%
País más cercano a Bitcoin en términos de consumo de electricidad.	Bangladesh
Electricidad estimada utilizada el día anterior (KWh)	149,143,384
Vatios implícitos por GH / s	0.118
Hashrate total de la red en PH / s (1,000,000 GH / s)	52,799
Electricidad consumida por transacción (KWh)	405
Número de hogares estadounidenses que podrían ser alimentados por Bitcoin	5,040,494
Número de hogares estadounidenses alimentados durante 1 día por la electricidad consumida para una sola transacción	13.69
Consumo de electricidad de Bitcoin como porcentaje del consumo de electricidad del mundo.	0.24%
Huella de carbono anual (kt de CO2)	25,858
Huella de carbono por transacción (kg de CO2)	192.41

Figura A.2.12: Datos de interés sobre el consumo energético de *Bitcoin*, de acuerdo con la metodología de [50].

En el caso de las *blockchains privadas*, en cambio, no se requiere de tanta potencia de cálculo ya que, como se ha abordado en ítems anteriores, las transacciones son validadas por un número acotado de usuarios (no existe competencia) y los *protocolos de consenso* se trabajan sistemáticamente para optimizar los recursos computacionales [15].

- La actualización y réplica de la *cadena de bloques* requiere de memoria computacional en constante crecimiento para almacenarse, lo que implica que se debe invertir en servidores cada vez más grandes para almacenar algunos datos (escalabilidad). Según la web "www.bitinfocharts.com", citada en [18], el tamaño de la *blockchain* de *Bitcoin* es de $235{,}86[GB]$ y la de *Ethereum* es cercana a $174{,}87[GB]$ (al 31 de enero de 2019).

- Dado que los *smart contracts* son inmutables, estos no pueden modificarse ante eventuales errores en su código. Además, debido a que los *contratos* trabajan exclusivamente con datos que viven en la *blockchain*, se imposibilita toda comunicación con aplicaciones externas (eso, hasta la llegada de *Oraclize*, como se explica en el Capítulo 5).

- Los avances en materia legal para la adecuada implementación de *blockchain* aún son insuficientes. Hace falta que el aparato jurídico entienda los puntos fuertes y débiles de esta tecnología a fin de explotar los beneficios en un marco regulatorio bien definido.

8.3 Anexos Capítulo 3

8.3.1 Información técnica del prototipo tecnológico de Blockchain GTIME

Ficha técnica *phiNergy*	
Datos generales	
Tensión de alimentación	5 V_{DC} ; 220 V_{AC}
Consumo	2,2 W | 5 V_{DC}
Temperatura de funcionamiento	{-40 °C ; 70 °C}
Sensor de corriente (monofásico)	10 A ; 30 A
Medición	
Resolución	12 bits
Frecuencia de muestreo	1 Hz
Exactitud	0,1 V_{DC}
Comunicación	
Registrador de datos	Raspberry Pi Zero W
Interfaz WLAN	802.11b 802.11g 802.11n | 2,4 GHz
Plataforma Web	
Tipo de almacenamiento de datos	MySQL
Frecuencia de actualización	Horaria

TablaA.3.1: Características técnicas medidor *phiNergy*, diseñado para medir en baja tensión [23].

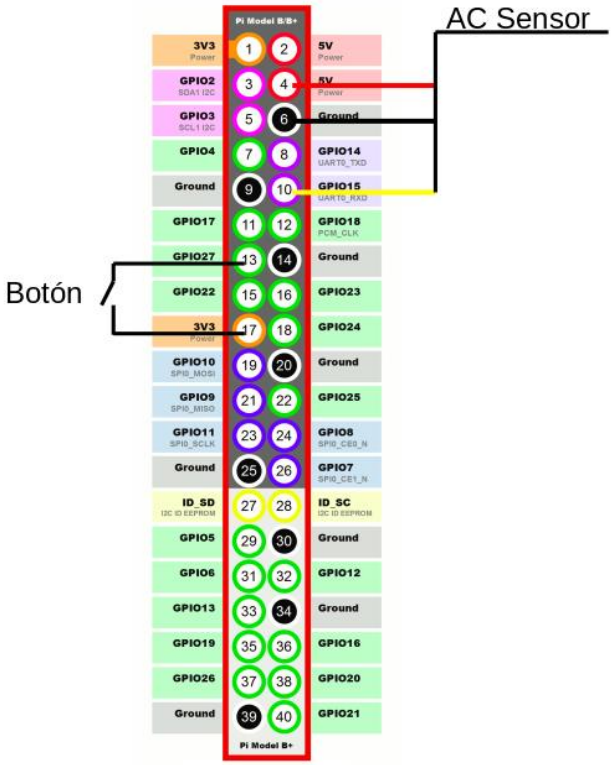


Figura A.3.1: Pines de conexión de la pinza amperimétrica y del interruptor en el GPIO de un medidor *phiNergy* [23].

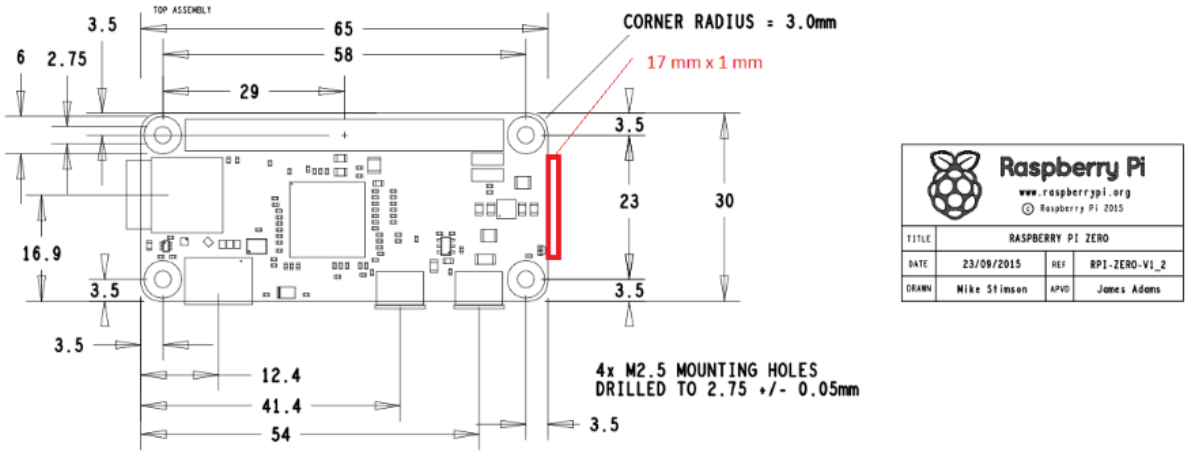


Figura A.3.2: Dimensiones de placa PCB utilizada en medidor *phiNergy* [35].

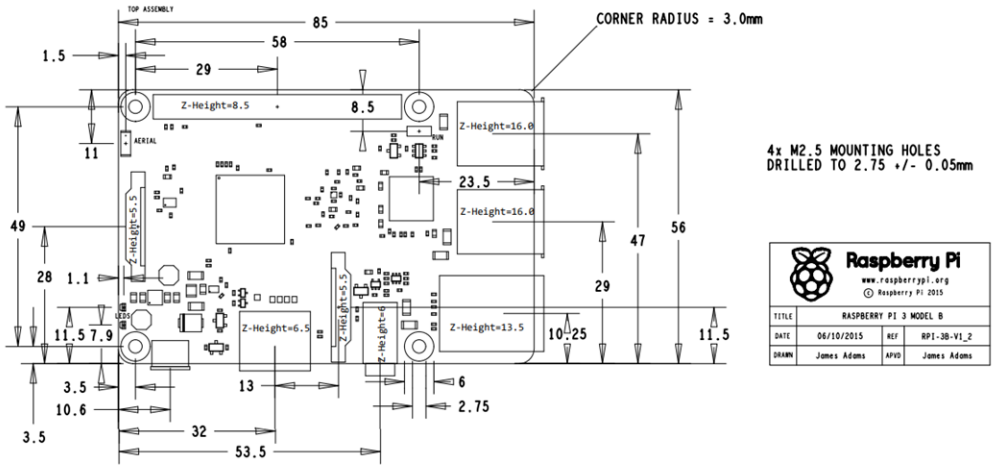


Figura A.3.6: Dimensiones de placa PCB utilizada en validador *Miner* [35].

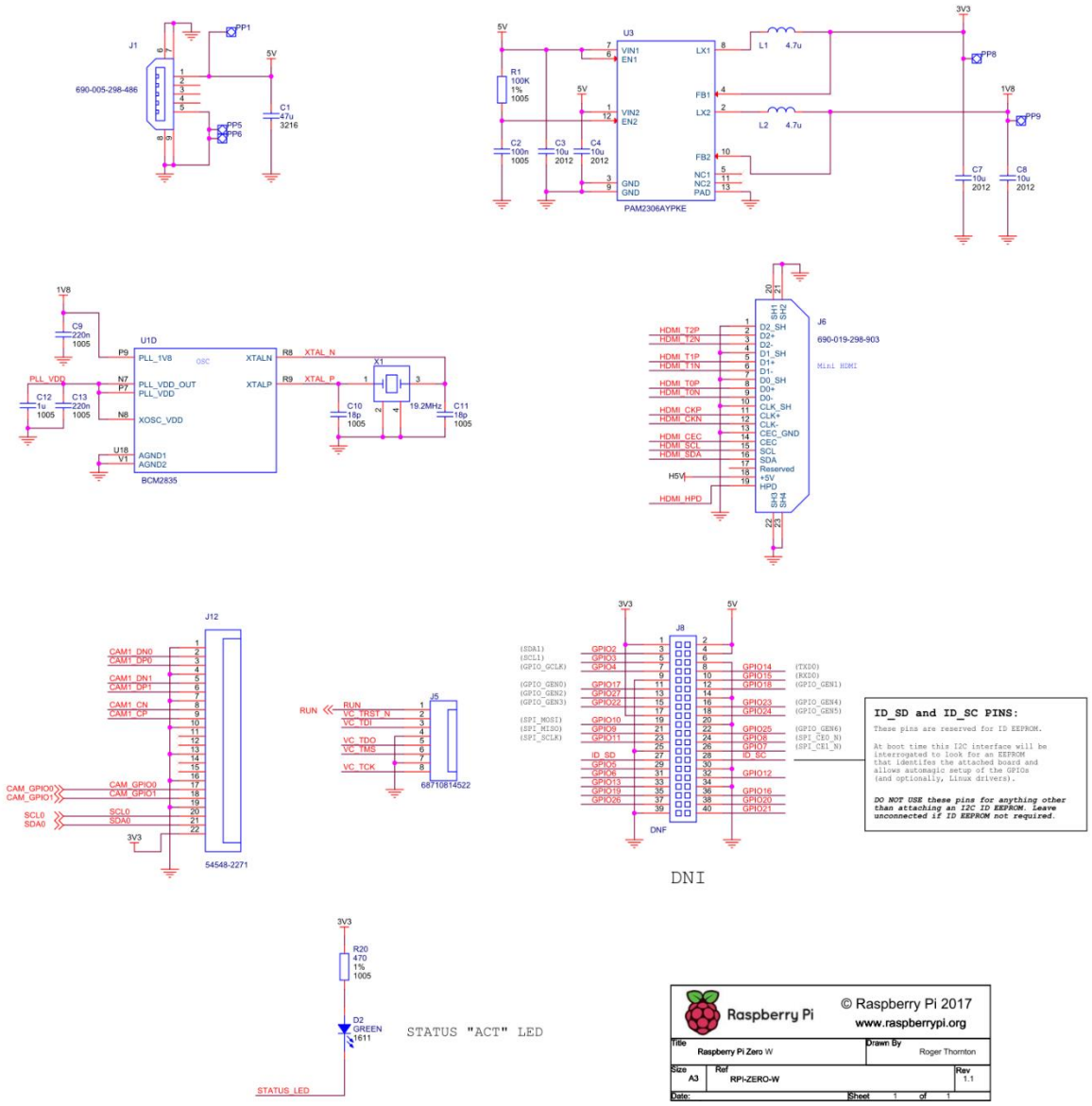


Figura A.3.3: Esquema electrónico oficial de una "Raspeberry Pi Zero W" empleada en *phiNergy* y publicado por [35].

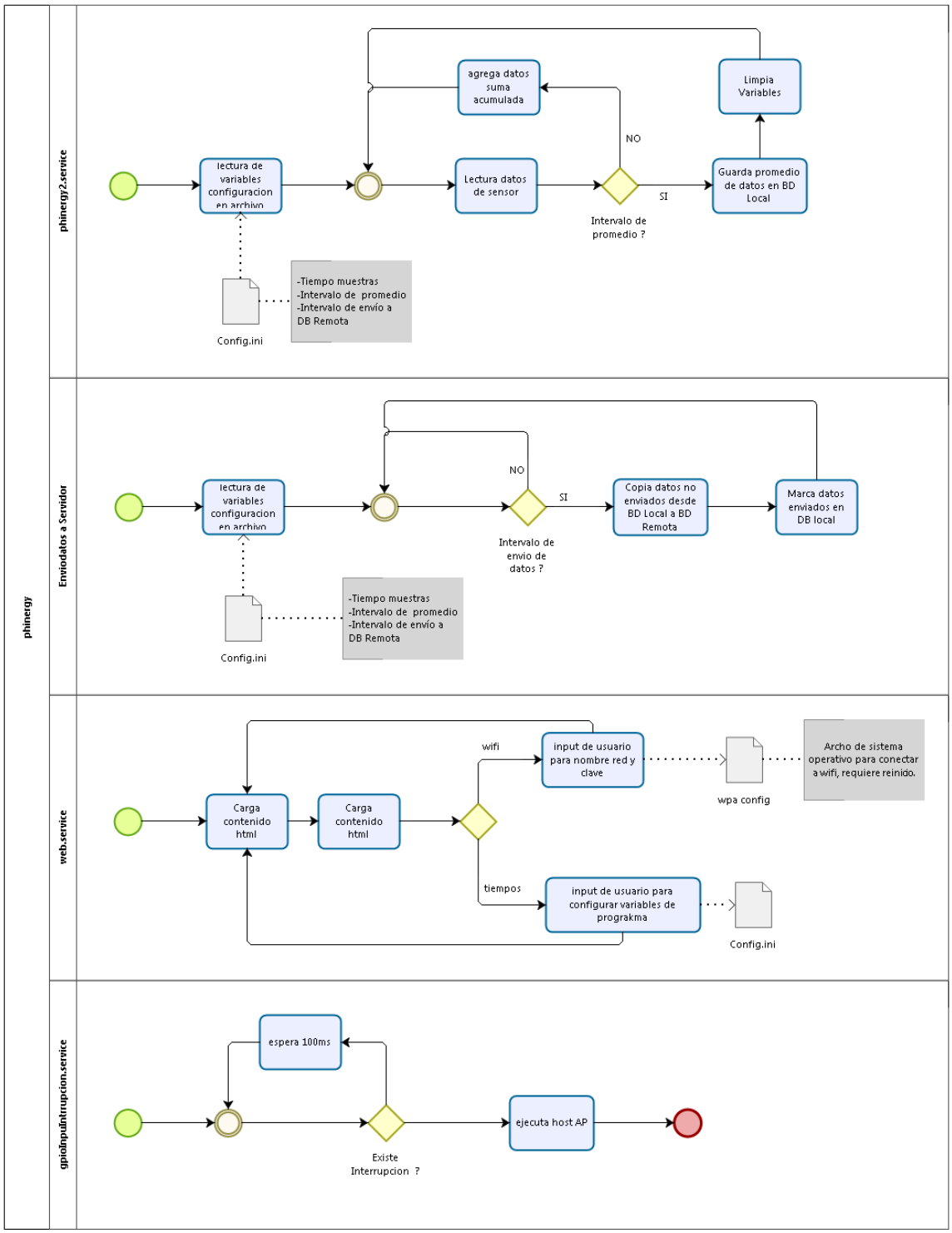


Figura A.3.4: Diagrama de bloques de las tareas programadas dentro de una *phiNergy* [23].

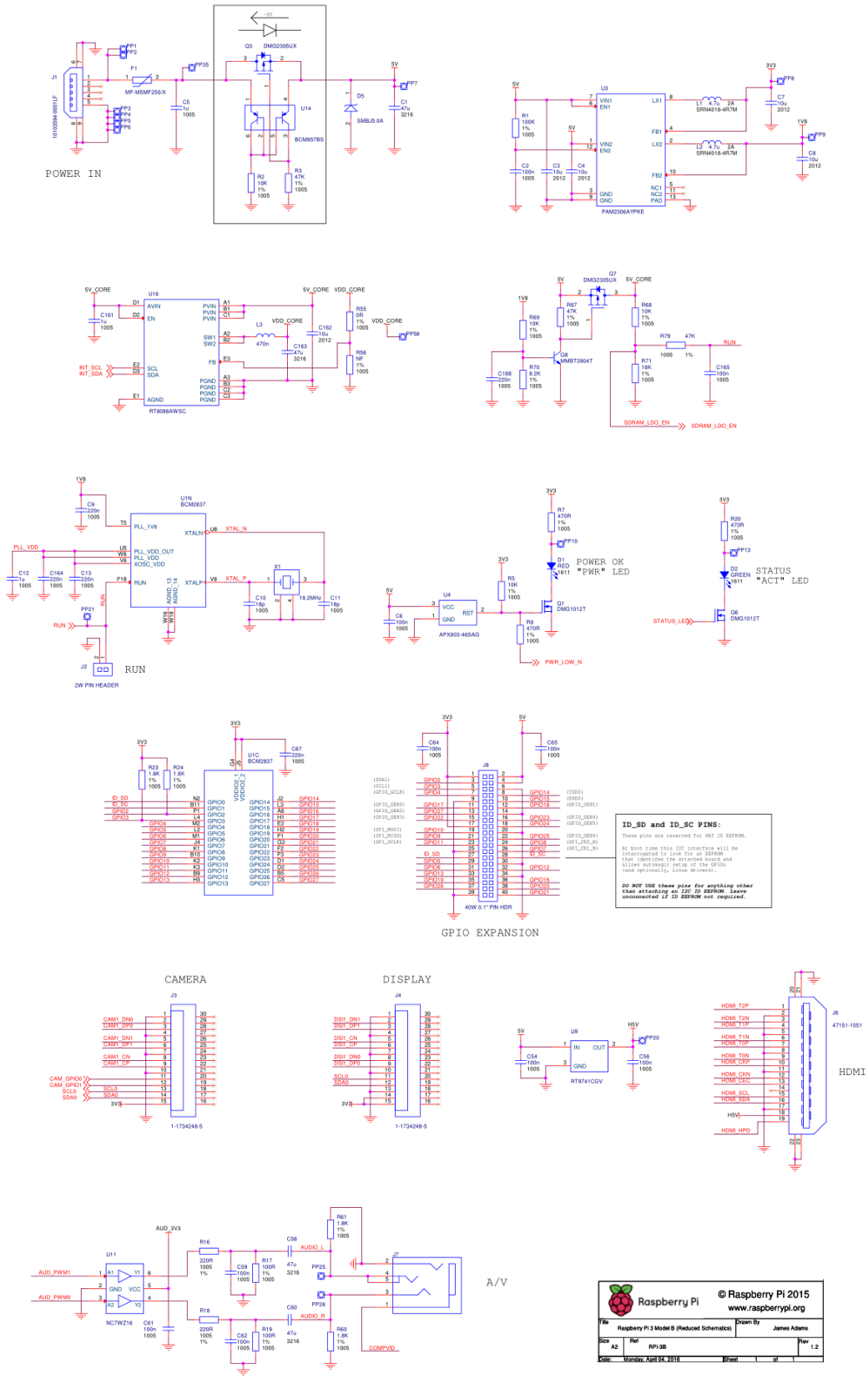


Figura A.3.5: Esquema electrónico de una "Raspeberry Pi 3 Modelo B", utilizada en *Miner* y publicado por [35].

8.3.2 Minería de datos, protocolo de consenso y escalabilidad del historial


```
blockChain.json ×
{
    "chain_SelloSol": [
        {
            "index": 0,
            "previousHash": "",
            "timestamp": "2018-06-27T13:16:29.631Z",
            "data": "Genesis block",
            "hash": "ec4e531141f75f930cddaf0ee3309b0bece54f2c345737270322e8b2e1548063",
            "nonce": 0,
            "createdBy": ""
        },
        {
            "index": 1,
            "previousHash": "ec4e531141f75f930cddaf0ee3309b0bece54f2c345737270322e8b2e1548063",
            "timestamp": "2018-06-27T13:49:30.883Z",
            "data": [
                [
                    {
                        "Geo": "-33.42914;-70.6225094",
                        "Time": "2018-06-27 13:32:32",
                        "Id": "a57f8d52d9d0d5ab18eeb3cc74f8e9c9e160dc1172e2a14d258655b0989df879;a57f8
                        "Mac": "b8:27:eb:ac:de:e8",
                        "Energy": 943.5347591999999
                    }
                ],
                [
```


Figura A.3.7: Bloque génesis de *Blockchain GTIME*. "Screenshot" del IDE Visual Studio Code en febrero de 2019 [23].

Como se ha dicho antes, la *minería de datos* bajo el protocolo *GTIME* está basada en el *protocolo de consenso PoW* y la función criptográfica es la SHA-256. Sin embargo, a diferencia de *Bitcoin*, por la naturaleza *hibrida* de *Blockchain GTIME*, no hay cabida para que se produzcan *bifurcaciones* o *forks*; por lo cual no se deben sumar esfuerzos computacionales para descubrir cuál es la versión "honesta" de la *cadena*, ya que existe una versión única (ante dudas, vaya al ítem 8.2.1).

La *Figura A.3.7* muestra el aspecto que tiene la *blockchain* de *Phineal* desde el entorno de desarrollo o "IDE de programación" llamado Visual Studio Code. Con dicho programa es posible observar las líneas de código que reúne el archivo alojado en el *Server*, llamado "`blockChain.json`", y que, en términos generales, es un arreglo o matriz que contiene a los vectores *GTIME*, que también son arreglos conformados por las cinco variables mencionadas en el ítem 3.1 del Capítulo 3.

Algo interesante de la imagen anterior es que se expone el inicio de la *cadena de bloques*. Se observa tanto el *bloque génesis* como el primer bloque añadido (*índex 1*), el cual se agregó el 27 de junio de 2018 a las $13:32:32\ [hrs]$ por la *phiNergy* de *MAC Address* $\mathrm{b8:27:eb:ac:de:e8}$, registrando $943{,}53[Wh]$ de energía en las coordenadas geográficas $\{-33{,}42914\ ;\ -70{,}6225094\}$.

En la *Figura A.3.7* también se debe notar que la *ID* del usuario del primer bloque está repetida en dos "palabras" hexadecimales de 64 dígitos cada una ($256[bits]$) y separadas por un " ; ", pero el segundo ejemplar no se muestra debido a su extensión. Esto se debe a que el protocolo *GTIME* está pensado para que en un futuro próximo se pueda albergar la identificación de los medidores emisor y receptor de la transferencia, respectivamente. Pero por ahora sigue estando en fase de prueba.

Respecto del tamaño de la *cadena*, si bien el almacenamiento de transacciones en el *Server* aún no presenta cifras críticas, a fines de enero de 2019 *Blockchain GTIME* tenía más de 15.000 bloques concatenados, traduciéndose en cerca de $17[MB]$ ocupados en la memoria del *Server*.

En virtud de ello, personal de la empresa [23] está estudiando la posibilidad de reducir el tamaño de la información guardada en los próximos bloques que se acoplen a la *blockchain*. Para no interferir en la calidad de los datos empíricos, una posibilidad que toma fuerza es acortar la *ID del usuario* mediante el cambio de la función criptográfica con la que se obtiene el cifrado alfanumérico de dicho parámetro; pasando de la SHA-256 a la SHA-0, es decir, usando tan sólo 40 dígitos de extensión para describir la *ID del usuario* ($160[bits]$).

En lo que sigue se expone el análisis planteado por Joaquim Quiroga Schmider, Ingeniero de Proyectos y Desarrollador en la empresa *Phineal*, donde se justifica porqué el tamaño de la *cadena de bloques* almacenada en memoria debe optimizarse; representando un punto de partida para mejorar la escalabilidad de registro de *Blockchain GTIME*.

Elementos básicos para conocer el tamaño de los bloques en *Blockchain GTIME*:

- Bloque base (b_b): representa el bloque sin contenido, pero con toda la "meta-data" necesaria para la construcción de un bloque. Tiene un tamaño promedio de $260\ [B]$ (Bytes) y cambia dependiendo del *índex*, el *nonce* y *nombre* del *Miner*. Un ejemplo es como sigue (notar que en la sección "data" no hay contenido alguno):


```
{"index":13774,

"previousHash":"000bbe3cc8971d933799a82d092842301216834dfebb85603eb06f3695183
62c",

"timestamp":"2019-01-16T16:37:08.673Z",

"data": [],

"hash":"00065b3a92ce9492a13277846eb1ab4962f28b1b983ba5ccb0812a0768f347f4",

"nonce":2330,

"createdBy":"Miner005"}
```

- Transacciones (tr_x): corresponde al conjunto de datos agrupados en un vector *GTIME*. Existe uno por cada registro de los nodos medidores. El tamaño del vector cambia con la cuantía de la energía medida y con la precisión de la geolocalización. Tiene un tamaño promedio de $245\ [B]$. Ejemplo de esto se muestra a continuación:


```
[{"Geo":"-33.428915; -70.621132",
"Time":"2019-01-16 15:17:14",
"Id":"94f2e7668dcf7b45c6291881b6bf2c962a20d0f71d52152f7a7e7160fa18e223;
94f2e7668dcf7b45c6291881b6bf2c962a20d0f71d52152f7a7e7160fa18e223",
"Mac":"40:9f:38:64:e7:e7",
"Energy":89317.5}]
```


- Cantidad de transacciones subidas por período de sellado ($N°_T$): actualmente se crea un bloque cada 20 minutos. Por otra parte, cada equipo de medición levanta una transacción (o un vector *GTIME*) cada 15 minutos. Por lo cual se puede decir que se suben 1,33 transacciones por cada período en que se sella un bloque.

- Nodos medidores activos en la red (phi_{med}): es la cantidad de equipos midiendo y participando del envío de datos con la *blockchain*.

Considerando los parámetros recientemente enunciados, el tamaño real de un bloque en la *Blockchain GTIME*, se puede calcular en base a la siguiente expresión:

$$Tamaño\ bloque \cong b_b + N°_T \cdot phi_{med} \cdot tr_x$$

Así, si hay mil nodos medidores subiendo información a un bloque vacío, se consigue:

$$\Rightarrow Tamaño\ bloque \cong 260[B] + 1{,}33\left[\frac{transacciones}{medidor}\right] \cdot 1.000[medidores] \cdot 245\left[\frac{B}{transacción}\right]$$

$$\therefore Tamaño\ bloque \cong 326.110[B] \approx 326{,}1[kB] \approx 0{,}326[MB]$$

Luego, dado que cada bloque se añade a la *cadena* en períodos de 20 minutos, es razonable pensar que se sellan aproximadamente 72 bloques por día, con lo cual el crecimiento diario de la *Blockchain GTIME* se estima en $23{,}47[MB]$, traduciéndose en $8.567{,}28[MB]$ al año. Esta información es importante si se considera que los nodos medidores de la red distribuida aumentarán de manera importante con los últimos proyectos adjudicados por la empresa. Con dicho nivel de crecimiento, la

capacidad de almacenamiento con la que cuenta el *Server* requiere especial atención, de manera que es pertinente estudiar la optimización de la *cadena de bloques* en función del propósito, las aplicaciones y las características de *Blockchain GTIME*.

8.3.3 Aplicaciones: Sello Sol, el blockchain solar de Latinoamérica

La empresa *Phineal* lleva más de tres años desarrollando *Sello Sol*, que es una certificación para distinguir a compañías e instituciones que realizan sus procesos empleando energía solar fotovoltaica y de concentración térmica [24]. Desde un principio, la idea ha sido que los consumidores finales tengan acceso a una plataforma web que garantice la utilización del recurso solar en la elaboración de productos y/o servicios mediante la publicación de datos relativos a la producción energética de las plantas solares utilizadas para ello, independientemente del tipo de abastecimiento con el que cuente la empresa (a saber, generación distribuida para autoconsumo o contrato de suministro con una generadora solar).

Figura A.3.8: Logo de *Sello Sol*, plataforma de certificación de uso de energía solar mediante *Blockchain GTIME* [25].

En un comienzo, *Sello Sol* se valió de los datos meteorológicos agrupados en la plataforma propietaria de información solar *phiNet*, la cual se sustenta del registro y actualización de datos empíricos de radiación solar, temperatura y velocidad del viento levantados por una red de estaciones manufacturadas por *Phineal* (*phiNet10/11*) y distribuidas a lo largo de todo Chile. Ahora en cambio, con la incorporación de *Blockchain GTIME* en su metodología, *Sello Sol* contará con una certificación auditable de la producción y el consumo medidos *in situ*, mostrando la cantidad de CO_2 que se evita arrojar a la atmósfera para fabricar un producto o llevar a cabo un servicio, respectivamente [25]. De este modo, se

espera fomentar la confección de bienes con más “atributos verdes”, así como de potenciar la diferenciación de productos y servicios gestados por medio de la energía solar.

Figura A.3.9: Infografía del servicio brindado por la certificación *Sello Sol* [24][25].

Así, en virtud de las características intrínsecas del *Blockchain GTIME* [26], *Sello Sol* vislumbra tres grandes oportunidades [25]:

- **Trazabilidad de energía con certificados de reducción de CO_2**
  Permite el monitoreo del historial de inyecciones de energía solar mediante plataforma *blockchain*, o bien, del abastecimiento de energía solar contemplando toda la *cadena de suministro* vía *smart contract*.

- **Contratos de suministro de compra/venta de energía solar basados en *blockchain***
  Consta de *smart PPAs* ejecutados por *smart contracts* que trabajan el balance energético, la valorización de las transferencias y el marco legal vigente sobre la red.

- **Transacciones de energía solar en sistemas de electromovilidad**
  Comprende la gestión energética basada en trazabilidad solar y en *smart contracts*, potenciando las recargas limpias y la venta de almacenamiento entre la red, los puntos de carga y los vehículos.

De este modo la iniciativa, canalizada a través de cualquiera de los servicios descritos, da pie a que las instituciones certificadas conozcan, en todo momento, la cantidad de energía producida por la planta con la que negociaron el *smart contract* y, en virtud de ello, puedan agregar valor a sus procesos industriales, comerciales u otros. También, los consumidores finales se verán directamente beneficiados, ya que podrán corroborar en tiempo real los datos de recurso primario y electricidad utilizados por las empresas u organizaciones para generar sus productos o servicios [23] [25].

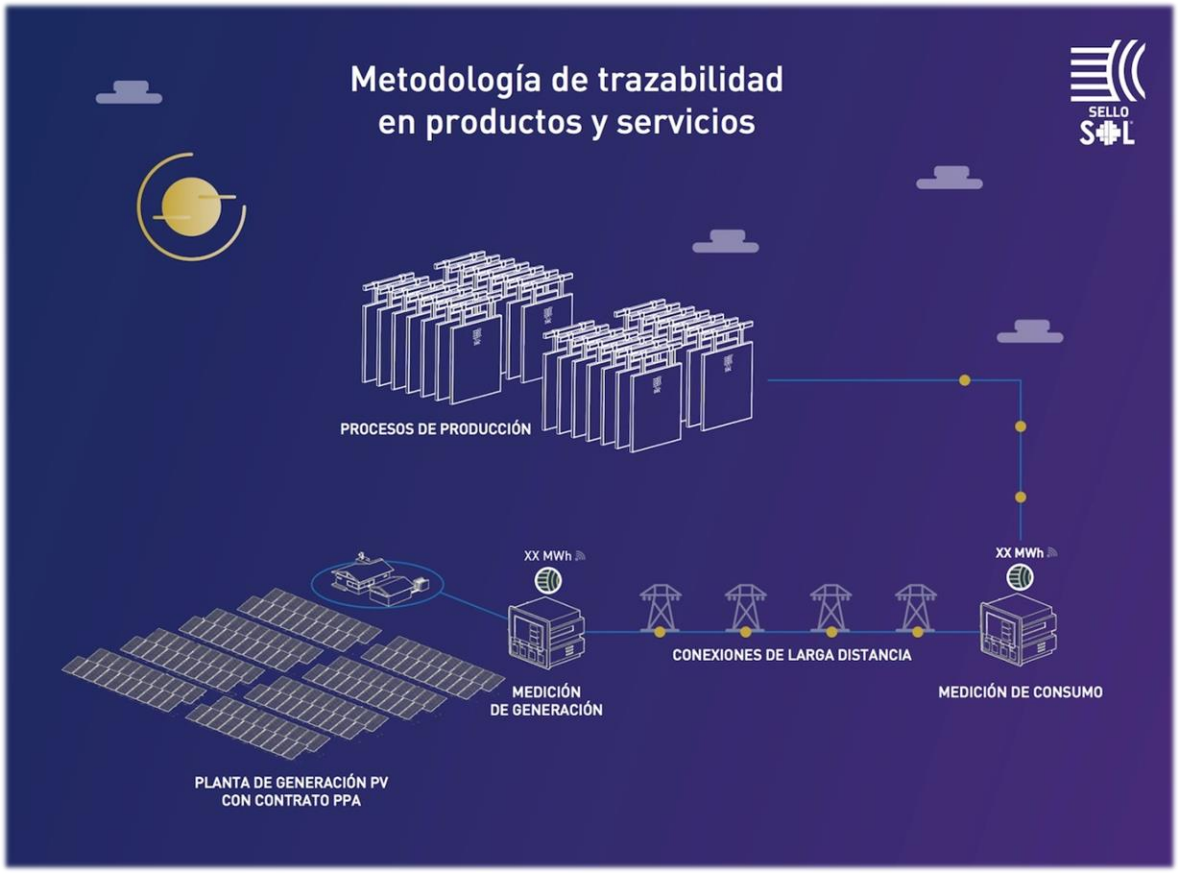


Figura A.3.10: Esquema de medición y captura de datos de energía eléctrica de acuerdo con la iniciativa *Sello Sol* [24].

8.4 Anexos Capítulo 4

8.4.1 Ejemplo del Método de Bialek usando Flujos Gruesos

El fin de utilizar *flujos gruesos* es repartir el total de pérdidas de la transferencia energética entre las cargas del sistema. Para conseguir esto, J. Bialek propone desplazar las pérdidas eléctricas al extremo receptor de las líneas de transporte. Por ejemplo, si se considera el SEP de 6 barras y 8 líneas de la *Figura A.4.1*, así como los datos de entrada de la *Tabla A.4.1* y los resultados del flujo de carga AC publicado por M. Chamba [3] en la *Tabla A.4.2*, se notará, por ejemplo, que para el tramo 1-4 existen $4{,}8[MW]$ asociados a *Efecto Joule*; debido a que se retiran $53{,}3[MW]$ desde la barra 1 y se inyectan $48{,}5[MW]$ en la barra 4. Estas pérdidas se pueden sumar al extremo receptor para conseguir un *flujo grueso* de $53{,}3[MW]$ a través de la línea de transmisión 1-4. Bajo este último procedimiento, es evidente que, si se replica la construcción de *flujos gruesos* para todas las líneas del SEP, la red de transporte no incorporará pérdidas y, por lo tanto, es admisible la aplicación del *algoritmo de inyecciones de potencia*.

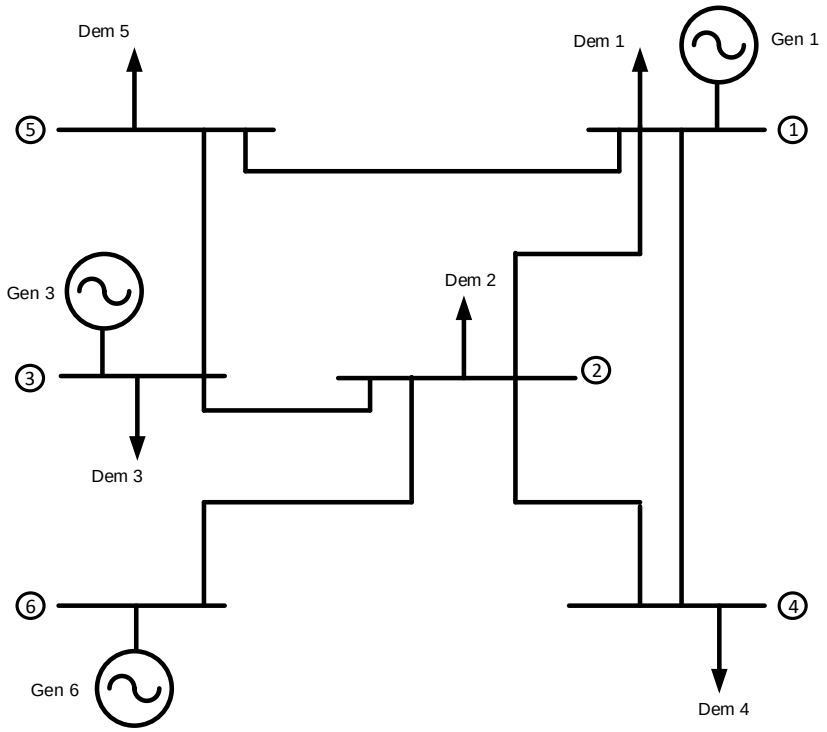


Figura A.4.1: Sistema de Garver utilizado por M. Chamba en su trabajo de título [3]. Elaboración propia.

Datos de entrada:

Barra	Generación [MW]	Demanda [MW]
1	191,57	56,00
2	0,00	168,00
3	255,67	28,00
4	0,00	112,00
5	0,00	168,00
6	126,00	0,00
Total [MW]	573,24	532,00
Pérdidas [MW]	41,24	

Tabla A.4.1: Datos de entrada publicados en [3].

Línea de Transmisión		Flujo de Potencia [MW]	
Barra "i"	Barra "j"	Pij_ret(i)	Pij_iny(j)
1	2	10,24	10,12
1	4	53,30	48,50
1	5	72,03	69,30
3	2	123,40	102,33
2	4	68,99	63,50
6	2	126,00	124,54
3	5	104,27	98,70
Total [MW]		558,23	516,99
Pérdidas [MW]		41,24	

Tabla A.4.2: Flujo de carga AC proporcionado por M. Chamba [3].

	Balance Nodal [MW]	
Barra	Inyección	Retiro
1	191,57	191,57
2	236,99	236,99
3	255,67	255,67
4	112,00	112,00
5	168,00	168,00
6	126,00	126,00

Tabla A.4.3: Sumatoria de inyecciones y retiros por barra. Elaboración propia.

Con lo anterior, definiendo la expresión para la *potencia nodal gruesa* P_i^{gruesa} que atraviesa el nodo *i* de un SEP de *n* barras en función de las inyecciones de potencia generada en dicho punto, se tiene:

$$P_i^{gruesa} = \sum_{j \in \alpha_i^{(u)}} \left|P_{i-j}^{grueso}\right| + P_{G_i} \quad ; \quad \forall\, i \in \{1, n\}$$

Hasta aquí, un ejemplo concreto alusivo a la red de la imagen sería notar que la *potencia nodal gruesa* expresada en función de los flujos inyectados en la *barra 4* está definida por la igualdad:

$$P_4^{gruesa} = P_{4-1}^{grueso} + P_{4-2}^{grueso} + P_{G_4} \Leftrightarrow P_4^{gruesa} = 53{,}30[MW] + 68{,}99[MW] + 0$$

$$\Leftrightarrow P_4^{gruesa} = 122{,}29[MW]$$

Luego, bajo el supuesto de que no existen pérdidas en el transporte, es posible escribir que $\left|P_{i-j}^{grueso}\right| \equiv \left|P_{j-i}^{grueso}\right|$ y, con ello, se puede afirmar que el factor $c_{ji}^{grueso} = \frac{\left|P_{j-i}^{grueso}\right|}{P_j^{gruesa}} = \frac{\left|P_{i-j}^{grueso}\right|}{P_j^{gruesa}}$. No obstante, debido a que en general las pérdidas reales por las líneas son pequeñas, el autor [1] estipula que se puede asumir que $\frac{\left|P_{j-i}^{grueso}\right|}{P_j^{gruesa}} \approx \frac{|P_{j-i}|}{P_j}$, es decir, c_{ji}^{grueso} podría escribirse en términos de los flujos reales y de las potencias nodales reales del sistema. Esto último representa el único supuesto del método y equivale a asumir que la distribución de los *flujos gruesos* en cualquier barra del SEP es idéntica a la repartición que tienen los flujos reales sobre el mismo.

Así, en coherencia al *algoritmo de inyecciones de potencia*¸ se formula:

$$P_i^{gruesa} - \sum_{j \in \alpha_i^{(u)}} c_{ji}^{grueso} \cdot P_j^{gruesa} = P_{G_i}$$

$$\Leftrightarrow \quad P_i^{gruesa} - \sum_{j \in \alpha_i^{(u)}} \frac{|P_{j-i}|}{P_j} \cdot P_j^{gruesa} = P_{G_i} \quad ; \quad \forall\, i \in \{1, n\}$$

Para el ejemplo, se tiene (para comprobar balance de potencias nodales ver *Tabla A.4.3*):

$$P_4^{gruesa} - \left(\frac{|P_{1-4}|}{P_1} \cdot P_1^{gruesa} + \frac{|P_{2-4}|}{P_2} \cdot P_2^{gruesa} \right) = P_{G_4}$$

$$\Leftrightarrow P_4^{gruesa} - \frac{|53{,}30|}{191{,}57} \cdot P_1^{gruesa} - \frac{|68{,}99|}{236{,}99} \cdot P_2^{gruesa} = 0$$

$$\Leftrightarrow P_4^{gruesa} - 0{,}2782 \cdot P_1^{gruesa} - 0{,}2911 \cdot P_2^{gruesa} = 0$$

Luego, es posible enunciar un sistema matricial de ecuaciones que contemple al vector incógnito de *potencias nodales gruesas* $\langle P_{nodal}^{gruesa} \rangle$, a la matriz de distribución de inyecciones de potencia $\langle A_{\mathcal{U}} \rangle$ creada a partir de los flujos reales del SEP y al vector de generación $\langle P_G \rangle$ que permanece invariante bajo esta resolución. Se consigue:

$$\langle A_{\mathcal{U}} \rangle \cdot \langle P_{nodal}^{gruesa} \rangle = \langle P_G \rangle$$

Tal como explicita el material bibliográfico [1], contando con las *potencias nodales gruesas*, es posible conocer la contribución de cada generador del sistema sobre los *flujos gruesos* que circulan por las líneas de transmisión, así como la participación que cada generador tiene sobre la *demanda gruesa* solicitada por las cargas del SEP. De este modo, para el tránsito de potencia por la red de transmisión se consigue:

$$\left|P_{i-l}^{grueso}\right| = \frac{\left|P_{i-l}^{grueso}\right|}{P_i^{gruesa}} \cdot P_i^{gruesa} \approx \frac{|P_{i-l}|}{P_i} \cdot P_i^{gruesa} = \frac{|P_{i-l}|}{P_i} \cdot \sum_{k=1}^{n} A_{\mathcal{U}_{ik}}^{-1} \cdot P_{G_k} \quad ; \quad \forall\, l \in \alpha_i^{(d)}$$

Mientras que para las cargas se obtiene:

$$P_{\mathcal{D}_i}^{gruesa} = \frac{P_{\mathcal{D}_i}^{gruesa}}{P_i^{gruesa}} \cdot P_i^{gruesa} \approx \frac{P_{\mathcal{D}_i}}{P_i} \cdot P_i^{gruesa} = \frac{P_{\mathcal{D}_i}}{P_i} \cdot \sum_{k=1}^{n} A_{\mathcal{U}_{ik}}^{-1} \cdot P_{G_k} \quad ; \quad \forall\, i \in \{1, n\}$$

La repartición de las pérdidas eléctricas para cada consumo del sistema, en tanto, vendrá definida por la expresión:

$$\mathcal{D}: \ \Delta P_{loss_i} = P_{\mathcal{D}_i}^{gruesa} - P_{\mathcal{D}_i}$$

Aquí vale la pena comentar que el valor de las *potencias nodales gruesas* no siempre se puede conseguir por mera inspección visual y que su cálculo en sistemas eléctricos más complejos no es trivial.

Si bien el SEP que se ilustra en la *Figura A.4.1* es reducido y responde a un sistema de prueba, se ha implementado un código en MATLAB para la obtención de los resultados que está disponible y puede consultarse en el ítem 8.4.2 de este capítulo. El archivo ".m", por su parte, ha sido entregado como material complementario a este documento.

Los datos que constituyen el sistema matricial, así como las consecuencias que derivan de la resolución de este, se detallan a continuación.

Formulación del problema:

$$\langle A_u \rangle \cdot \langle P_{nodal}^{gruesa} \rangle = \langle P_G \rangle$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -0{,}0535 & 1 & -0{,}4827 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -0{,}2782 & -0{,}2911 & 0 & 1 & 0 & 0 \\ -0{,}3760 & 0 & -0{,}4078 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} P_1^{gruesa} \\ P_2^{gruesa} \\ P_3^{gruesa} \\ P_4^{gruesa} \\ P_5^{gruesa} \\ P_6^{gruesa} \end{pmatrix} = \begin{pmatrix} 191{,}57 \\ 0 \\ 255{,}67 \\ 0 \\ 0 \\ 126{,}00 \end{pmatrix}$$

Datos de salida:

Barra	Potencia Nodal Gruesa [MW]	Demanda Gruesa [MW]	Contribuciones de cada Generador Grueso sobre Demanda Gruesa [MW]					
1	191,57	56,00	56,00	0,00	0,00	0,00	0,00	0,00
2	259,64	184,06	7,26	0,00	87,48	0,00	0,00	89,32
3	255,67	28,00	0,00	0,00	28,00	0,00	0,00	0,00
4	128,88	128,88	56,28	0,00	35,92	0,00	0,00	36,68
5	176,30	176,30	72,03	0,00	104,27	0,00	0,00	0,00
6	126,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Total [MW]	-	573,24	Gen 1 [MW]	-	Gen 3 [MW]	-	-	Gen 6 [MW]
			191,57		255,67			126,00

Tabla A.4.4: Resultado de Flujos Gruesos sobre el Sistema de Garver de la *Figura A.4.1*. Elaboración propia.

Asignación de pérdidas a consumidores	
Total Demanda Gruesa [MW]	573,24
Total Demanda Real [MW]	532,00
Pérdidas [MW]	41,24

Tabla A.4.5: Pérdidas totales del SEP de prueba según Flujos Gruesos. Elaboración propia.

De la *Tabla A.4.4* se observa que la *potencia nodal gruesa* es consistente al supuesto planteado en el Método de Bialek vía Flujos Gruesos, por cuanto el balance de energía nodal está compuesto por la sumatoria de las inyecciones o retiros de potencias reales en cada barra en conformidad al *principio de conservación de la energía*.

Por ejemplo, en la *barra 1*, se puede escribir en términos de la generación real inyectada (que no varía según esta resolución), como:

$$P_1^{gruesa} = P_{G_1} \cong 191{,}57[MW]$$

O bien, en función de los retiros reales efectuados sobre dicha barra (equivalentes a los *flujos gruesos* que se surten desde la *barra 1*), según:

$$P_1^{gruesa} = P_{1-2}^{grueso} + P_{1-4}^{grueso} + P_{1-5}^{grueso} + P_{\mathcal{D}_1}^{gruesa}$$

$$\Leftrightarrow P_1^{gruesa} = 10{,}24 + 53{,}30 + 72{,}03 + 56{,}00 \cong 191{,}57[MW]$$

Por otro lado, la *Tabla A.4.4* también exhibe los aportes de potencia que hace el parque de generación a los flujos solicitados por la demanda, permitiendo corroborar qué generadores están alimentando a una carga particular. Al respecto es interesante notar que, en este ejemplo, los generadores de mayor producción suministran potencia a casi la totalidad de los consumos del SEP de la *Figura A.4.1*, a diferencia de G_6, que sólo contribuye con dos cargas. Al mismo tiempo, se aprecia que las *demandas gruesas* más grandes se abastecen, en general, con aportes de todos los generadores (por ejemplo, $\mathcal{D}_2$ y $\mathcal{D}_4$). La excepción la refleja $\mathcal{D}_5$, que, a pesar de ser la segunda carga con más retiro de potencia, no se alimenta de G_6; justificando que es más "eficiente" alimentarse desde las líneas 1-5 y 3-5, en lugar de un suministro procedente desde las líneas 1-5, 3-5 y 5-6.

Por otra parte, dado que se han desplazado las pérdidas al bloque de demanda, es posible afirmar que la suma de las porciones de potencia retiradas desde un generador puntual es igual a la potencia inyectada por dicho generador. Por ejemplo, para G_3 se aprecia que los retiros efectuados por las *demandas gruesas* $\mathcal{D}_2$, $\mathcal{D}_3$, $\mathcal{D}_4$ y $\mathcal{D}_5$ determinan el total del suministro inyectado desde el *nodo 3*. En efecto:

$$\frac{P_{\mathcal{D}_2}}{P_{nodal_2}} \cdot A_{u_{26}}^{-1} \cdot P_{G_6} + \frac{P_{\mathcal{D}_4}}{P_{nodal_4}} \cdot A_{u_{46}}^{-1} \cdot P_{G_6} = 89{,}32 + 36{,}68 \cong 126{,}00[MW] = P_{G_6}$$

Sumado a lo anterior, se puede inferir que las contribuciones sobre los flujos de las cargas son mayores en función de la cargabilidad de las líneas (cantidades de potencia que se pueden transmitir, entregadas por "default" en este sistema de prueba), de los niveles de demanda y de la existencia de un camino físico a través de la red que permita "unir" el punto de suministro con el punto de consumo. Un ejemplo claro es lo que sucede con G_3, quien llega a cubrir la totalidad de la demanda de $\mathcal{D}_3$ además de sostener importantes aportes sobre los flujos demandados por $\mathcal{D}_2$ y $\mathcal{D}_5$, respectivamente. Adicionalmente, conviene advertir que dicha conexión directa no necesariamente debe ser inmediata, es decir, puede estar compuesta por más de un tramo de líneas; permitiendo que el flujo atraviese distintas subestaciones para concretar el suministro energético (como en un sistema radial, por ejemplo).

Finalmente, en comparación a los datos de entrada expuestos en la *Tabla A.4.1*, se observa que la magnitud de las *demandas gruesas* es mayor o igual a los valores de las demandas reales del SEP, debido a que se traspasan las pérdidas eléctricas a los consumidores del sistema. Esto último se explicita en la *Tabla A.4.5* que muestra la diferencia entre ambas "demandas totales" y permite corroborar que, tal como se ha mencionado al inicio de este capítulo, la repartición del cargo físico por uso de la red de transmisión es una operación posterior que no afecta a la operación del SEP [5]. Para conseguir la cantidad exacta de pérdidas destinadas a cada demanda, sólo se debe hacer la diferencia entre la cuantía *gruesa* y real de cada carga del sistema [1]. Una vez conocida dicha cantidad, es posible despejar, mediante un prorrateo por nivel de participación en el flujo demandado, la cantidad exacta que cada generador aporta para conformar dichas pérdidas en la carga.

8.4.2 Código MATLAB Ej. Sistema de Garver – M. de Bialek Flujos Gruesos

```
%Ejemplo Bialek 6 barras "Flujos Gruesos"

%------------------------------Lectura de los datos del SEP------------------------------
%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
hojactiva=actarch.Activesheet;
Range=get(hojactiva,'Range','D3','E8');
%Matriz de array para GD
P1=Range.value;
invoke(oparch,'Close');
invoke(actarch,'Quit');
%Matriz numérica para Gen y Dem
[M N]=size(P1);
inarch=cat(2,P1(:));
GD=reshape(inarch,M,N);
Gen=GD(:,1);
Dem=GD(:,2);

%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
hojactiva=actarch.Activesheet;
Range=get(hojactiva,'Range','G4','J10');
%Matriz de array para Red
P2=Range.value;
invoke(oparch,'Close');
invoke(actarch,'Quit');
%Matriz numérica para Red
[H I]=size(P2);
inarch=cat(2,P2(:));
Red=reshape(inarch,H,I);

Lineas=[];
Flujos=[];

for i=1:size(Red,1)
    for j=1:size(Red,2)
        if j<3
```

```
        Lineas(i,j)=Red{i,j};
        elseif j>=3
        Flujos(i,j-2)=Red{i,j};
        end
    end
end

Nodoret=Lineas(:,1);
Nodoiny=Lineas(:,2);

%-----------------------------Construcción matrices de entrada---------------------------

P_G=[]; %muestra la matriz de generación del SEP "[P_G]"
for i=1:size(Gen,1)
    for j=1:size(Gen,1)
        if i==j
        P_G(i,j)=Gen{i,1};
        else
        P_G(i,j)=0;
        end
    end
end


P_D=[];%muestra la matriz de demandas de cada barra del SEP "[P_D]"
for i=1:size(Dem,1)
    for j=1:size(Dem,1)
        if i==j
        P_D(i,j)=Dem{i,1};
        else
        P_D(i,j)=0;
        end
    end
end


P_ret_barra1=[]; % matriz de flujos retirados desde la barra "fila" por la barra "columna".

for i=1:size(Gen,1)
    for j=1:size(Gen,1)

        for k=1:size(Lineas,1)
            for l=1:size(Lineas,2)

                if l==1
                    if i==Lineas(k,l)&&j==Lineas(k,l+1)

                        if Flujos(k,l)>Flujos(k,l+1)
                        P_ret_barra1(i,j)=Flujos(k,l);
                       elseif Flujos(k,l+1)>Flujos(k,l)
                        P_ret_barra1(i,j)=Flujos(k,l+1);
                        end
                elseif l==size(Lineas,2)
                    if i==Lineas(k,l-1)&&j==Lineas(k,l)

                        if Flujos(k,l-1)>Flujos(k,l)
                        P_ret_barra1(i,j)=Flujos(k,l-1);
                       elseif Flujos(k,l)>Flujos(k,l-1)
                        P_ret_barra1(i,j)=Flujos(k,l);
                        end
                    end
                    end
                end
            end
        end
    end
end

P_ret_barra=cat(2,P_ret_barra1,zeros(6,1));
```

```
P_iny_barra1=[]; %matriz de flujos inyectados en la barra "columna" por la barra "fila".

for i=1:size(Gen,1)
    for j=1:size(Gen,1)

        for k=1:size(Lineas,1)
            for l=1:size(Lineas,2)

                if l==1
                    if i==Lineas(k,l)&&j==Lineas(k,l+1)

                        if Flujos(k,l)<Flujos(k,l+1)
                        P_iny_barra1(i,j)=Flujos(k,l);
                       elseif Flujos(k,l+1)<Flujos(k,l)
                        P_iny_barra1(i,j)=Flujos(k,l+1);
                        end

                elseif l==size(Lineas,2)
                    if i==Lineas(k,l-1)&&j==Lineas(k,l)

                        if Flujos(k,l-1)<Flujos(k,l)
                        P_iny_barra1(i,j)=Flujos(k,l-1);
                       elseif Flujos(k,l)<Flujos(k,l-1)
                        P_iny_barra1(i,j)=Flujos(k,l);
                        end

                    end
                    end
                end
            end
        end
    end
end

P_iny_barra=cat(2,P_iny_barra1,zeros(6,1));


Pnodo_G=zeros(size(Gen,1),size(Gen,1)); %matriz que agrupa las potencias totales a través de
                                          los nodos de generación

for i=1:size(Gen,1) %Filas matriz [Pnodo_G]
    for j=1:size(Gen,1) %Columnas matriz [Pnodo_G]
        if i==j&&i==1
            Pnodo_G(i,j)=P_G(i,j)+sum(P_iny_barra(:,j));
        elseif i==j&&i==3
            Pnodo_G(i,j)=P_G(i,j)+sum(P_iny_barra(:,j));
        elseif i==j&&i==6
            Pnodo_G(i,j)=P_G(i,j)+sum(P_iny_barra(:,j));
        end
    end
end


Pnodo_D=zeros(size(Gen,1),size(Gen,1)); %matriz que agrupa las potencias totales a través de
                                          los nodos de demanda

for i=1:size(Gen,1) %Filas matriz [Pnodo_D]
    for j=1:size(Gen,1) %Columnas matriz [Pnodo_D]
        if i==j&&i==2
            Pnodo_D(i,j)=P_D(i,j)+sum(P_ret_barra(i,:));
        elseif i==j&&i==4
            Pnodo_D(i,j)=P_D(i,j)+sum(P_ret_barra(i,:));
        elseif i==j&&i==5
            Pnodo_D(i,j)=P_D(i,j)+sum(P_ret_barra(i,:));
        end
    end
end


P_nodal_total=Pnodo_G+Pnodo_D; %muestra la potencia nodal total a través de cada barra del
                                    SEP bajo Flujos Gruesos
```

```
% ------------------------------Fin matrices de entrada----------------------------------

%-----------------------------Construcción de la matriz "[Au]"----------------------------

Au=[]; %matriz de coeficientes "[Au]", a rellenar
for i=1:size(Gen,1) %Filas matriz [Au]
    for j=1:size(Gen,1) %Columnas matriz [Au]
        if  j==i
            Au(i,j)=1;
        else
            Au(i,j)=-abs(P_ret_barra(j,i))/P_nodal_total(j,j);
        end
    end
end


% ---------------------------Fin construcción de la matriz "[Au]"-------------------------


Au %muestra la matriz de coeficientes "[Au]"

Au_inv=Au^(-1) %muestra la inversa de la matriz de coeficientes, denotada por "[Au]^(-1)"

PG=zeros(size(P_G,1),1);
for i=1:size(P_G,1)
    PG(i,1)=P_G(i,i);
end

PG % muestra el vector de generación "[PG]"

% ------------------------------Despeje de demandas gruesas-------------------------------

P_nodal_gruesa=(Au^(-1))*PG

% % ----------------------------Fin de despeje de demandas gruesas-------------------------

% % --------------------------Resultados Flujos Gruesos J.Bialek--------------------------

Pcarga_gruesa=[];
for i=1:size(Dem,1)
    Pcarga_gruesa(i,1)=(Dem{i,1}/P_nodal_total(i,i))*(P_nodal_gruesa(i,1));
end

Pcarga_gruesa %Muestra la potencia demandada en las cargas bajo el uso de factores gruesos

Aportes_G=[];
for i=1:size(Au_inv,1)
    for j=1:size(Au_inv,2)
        Aportes_G(i,j)=(Dem{i,1}/P_nodal_total(i,i))*Au_inv(i,j)*PG(j,1);
    end
end

Aportes_G %Muestra los aportes de energía del Generador "columna" en la demanda gruesa de la
               Carga "fila"


%-------------------------Registro de los resultados del Método de Bialek-------------------
%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
set(actarch,'Visible',1);%Activa la hoja de Excel
hojactiva=actarch.Activesheet;
hojaRange=get(hojactiva,'Range','D18','D23');
hojaRange1=get(hojactiva,'Range','E18','E23');
hojaRange2=get(hojactiva,'Range','F18','K23');
set(hojaRange,'Value',P_nodal_gruesa);%Escribir Factores Gruesos sobre hoja Excel
set(hojaRange1,'Value',Pcarga_gruesa);%Escribir Demandas Gruesas sobre hoja Excel
set(hojaRange2,'Value',Aportes_G);%Escribir las contribuciones de flujo de los Generadores
                                    sobre hoja Excel
```

```
invoke(oparch,'Close');
invoke(actarch,'Quit');
```

8.4.3 Ejemplo del Método de Bialek usando Flujos Medios

Si ahora se desea describir el tránsito de potencia bajo el concepto de *flujos medios*, entonces se deberá compartir el total de pérdidas de la transferencia energética entre generadores y demandas del sistema. Para lograr esto, el autor [1] plantea que se debe obtener el valor promedio del flujo de potencia que circula por una línea de transmisión para posteriormente asignar la mitad de las pérdidas a cada extremo de la línea. Esto último implica que los generadores conectados a la línea percibirán una disminución de sus inyecciones y que las cargas conectadas a la línea de transporte experimentarán un aumento en sus retiros. Dicho de otro modo, se resta la mitad de las pérdidas de cada línea a los generadores vinculados y se suma la mitad restante a las cargas relacionadas con dicha línea.

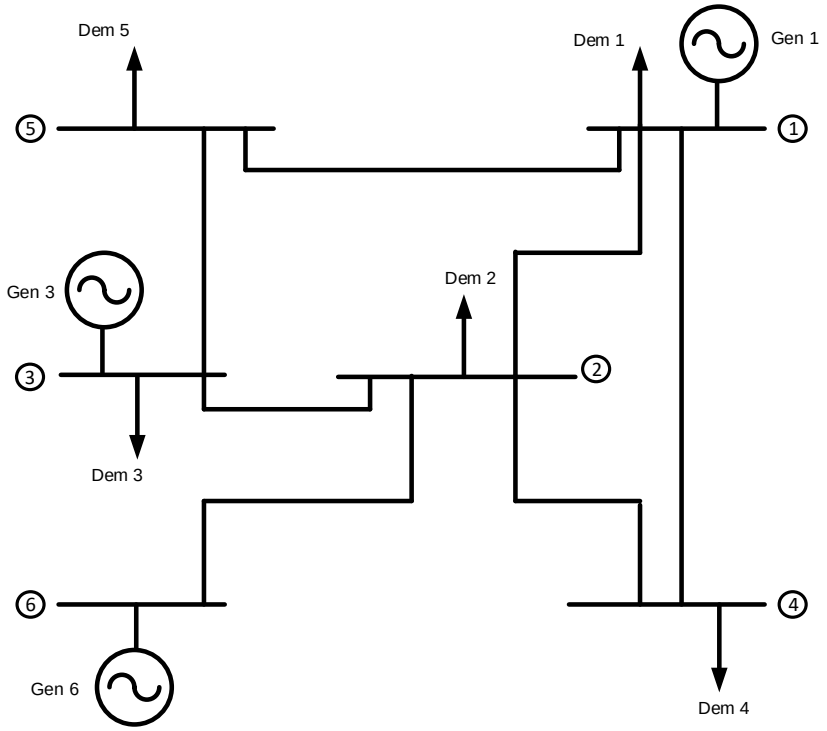


Figura A.4.2: SEP utilizado en [3] para ejemplificar las liquidaciones comerciales en Ecuador. Elaboración propia.

Al igual que en el ítem anterior, para el tramo 1-4 del sistema de prueba de la *Figura A.4.2* (que es la misma red de la *Figura A.4.1*, pero se ha repetido por comodidad) se aprecian $4{,}8[MW]$ asociados a *Efecto Joule* (ver *Tabla A.4.6* y *Tabla A.4.7*). Estas pérdidas se pueden dividir entre el extremo emisor y receptor para conceder $2{,}4[MW]$ a cada uno. Al mismo tiempo, el *flujo medio* que circula a través del tramo 1-4 sería igual a $50{,}9[MW]$, desligándose completamente de las pérdidas por concepto de transporte. Si este procedimiento se replica sobre todas las líneas del SEP, el sistema de transmisión carecería de pérdidas eléctricas y, con ello, el modelo podría resolverse mediante el *algoritmo de inyecciones de potencia* o el *algoritmo de retiros de potencia*. Para este ejemplo se aplicará el primero de los algoritmos mencionados.

Datos de entrada:

Barra	Generación [MW]	Demanda [MW]
1	191,57	56,00
2	0,00	168,00
3	255,67	28,00
4	0,00	112,00
5	0,00	168,00
6	126,00	0,00
Total [MW]	573,24	532,00
Pérdidas [MW]	41,24	

Tabla A.4.6: Estado de productores y consumidores para el Sistema de Garver de la *Figura A.4.2* [3].

Línea de Transmisión		Flujo de Potencia [MW]		
Barra "i"	Barra "j"	Pij_ret(i)	Pij_iny(j)	Pij_media
1	2	10,24	10,12	10,18
1	4	53,30	48,50	50,90
1	5	72,03	69,30	70,67
3	2	123,40	102,33	112,87
2	4	68,99	63,50	66,25
6	2	126,00	124,54	125,27
3	5	104,27	98,70	101,49
Total [MW]		558,23	516,99	
Pérdidas [MW]		41,24		

Tabla A.4.7: Flujo de carga AC proporcionado por M. Chamba [3].

	Balance Nodal [MW]	
Barra	Inyección	Retiro
1	189,66	189,66
2	248,32	248,32
3	249,01	249,01
4	117,15	117,15
5	172,15	172,15
6	125,27	125,27

Tabla A.4.8: Sumatoria de inyecciones y retiros por barra. Elaboración propia.

En virtud de lo expuesto, puede escribirse la *potencia nodal media* P_i^{media} que traspasa la barra *i* de un SEP de *n* barras en función de la potencia generada en dicha barra como:

$$P_i^{media} = \sum_{j \in \alpha_i^{(u)}} \left|P_{i-j}^{media}\right| + P_{G_i}^{media} \quad ; \quad \forall\, i \in \{1, n\}$$

En ausencia de pérdidas en la etapa de transmisión, es válido afirmar que $\left|P_{i-j}^{medio}\right| \equiv \left|P_{j-i}^{medio}\right|$ y, por ejemplo, se aprecia que la *potencia nodal media* descrita en términos de las potencias inyectadas en la *barra 4* viene definida por:

$$P_4^{media} = P_{4-1}^{media} + P_{4-2}^{media} + P_{G_4}^{media}$$

$$\Leftrightarrow\ P_4^{gruesa} = 50{,}90[MW] + 66{,}25[MW] + 0 = 117{,}15[MW]$$

Siguiendo con el método, la suposición anterior permite expresar que $c_{ji}^{medio} = \frac{\left|P_{j-i}^{medio}\right|}{P_j^{media}} = \frac{\left|P_{i-j}^{medio}\right|}{P_j^{media}}$ y, en coherencia al *algoritmo de inyecciones de potencia*¸ la relación entre la *potencia nodal media*, los *flujos medios* por las líneas y la *generación media* se puede escribir como sigue:

$$P_i^{media} - \sum_{j \in \alpha_i^{(u)}} c_{ji}^{medio} \cdot P_j^{media} = P_{G_i}^{media}$$

En este caso el ejemplo queda (para estudiar lo que pasa con las potencias nodales ver *Tabla A.4.8*):

$$P_4^{media} - \left(\frac{\left|P_{1-4}^{medio}\right|}{P_1} \cdot P_1^{media} + \frac{\left|P_{2-4}^{medio}\right|}{P_2} \cdot P_2^{media}\right) = P_{G_4}^{media}$$

$$\Leftrightarrow P_4^{media} - \frac{|50{,}90|}{189{,}66} \cdot P_1^{media} - \frac{|66{,}25|}{248{,}32} \cdot P_2^{media} = 0$$

$$\Leftrightarrow P_4^{media} - 0{,}2684 \cdot P_1^{media} - 0{,}2668 \cdot P_2^{media} = 0$$

Para descubrir el contenido del vector de *potencias nodales medias* $\langle P_{nodal}^{media} \rangle$ se construye un sistema matricial que reúne a la matriz de distribución de inyecciones de *flujos medios* $\langle A_u^{media} \rangle$ y al vector de *generación media* $\langle P_G^{media} \rangle$. Este último resulta de la modificación del vector de generación real como consecuencia de la asignación de pérdidas medias de transporte.

El sistema queda:

$$\langle A_u^{media} \rangle \cdot \langle P_{nodal}^{gruesa} \rangle = \langle P_G^{media} \rangle$$

En conocimiento de las *potencias nodales medias*, es posible conocer el aporte del parque generador sobre los *flujos medios* y las *demandas medias* del SEP. En efecto, para los flujos de potencia que circulan por la red de transmisión, se tiene:

$$\left| P_{i-l}^{medio} \right| = \frac{\left| P_{i-l}^{medio} \right|}{P_i^{media}} \cdot \sum_{k=1}^{n} A_{u_{ik}}^{media^{-1}} \cdot P_{G_k}^{media} \quad ; \quad \forall\, l \in \alpha_i^{(d)}$$

Mientras que para las cargas se obtiene:

$$P_{\mathcal{D}_i}^{media} = \frac{P_{\mathcal{D}_i}^{media}}{P_i^{media}} \cdot \sum_{k=1}^{n} A_{u_{ik}}^{media^{-1}} \cdot P_{G_k}^{media} \quad ; \quad \forall\, i \in \{1, n\}$$

Al terminar la aplicación del Método de Bialek usando Flujos Medios, el reparto de las pérdidas eléctricas para cada generador y demanda del sistema, está definido por:

$$G: \;\; \Delta P_{loss_i} = P_{G_i} - P_{G_i}^{media}$$

$$\mathcal{D}: \;\; \Delta P_{loss_i} = P_{\mathcal{D}_i}^{media} - P_{\mathcal{D}_i}$$

El SEP de la *Figura A.4.2* y la información expuesta tiene como fin ejemplificar el procedimiento de *flujos medios*, para lo cual se ha realizado un código en "lenguaje M" para la obtención de las *potencias nodales medias* y sus implicancias en los flujos requeridos por las cargas (ver ítem 8.4.4). El archivo ".m", en tanto, ha sido entregado como material complementario a este documento.

La enunciación del problema y los correspondientes datos de salida se detallan a continuación:

<u>Formulación del problema:</u>

$$\langle A_u \rangle \cdot \langle P_{nodal}^{media} \rangle = \langle P_G \rangle$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -0{,}0537 & 1 & -0{,}4533 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -0{,}2684 & -0{,}2668 & 0 & 1 & 0 & 0 \\ -0{,}3726 & 0 & -0{,}4076 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} P_1^{media} \\ P_2^{media} \\ P_3^{media} \\ P_4^{media} \\ P_5^{media} \\ P_6^{media} \end{pmatrix} = \begin{pmatrix} 189{,}66 \\ 0 \\ 249{,}01 \\ 0 \\ 0 \\ 125{,}27 \end{pmatrix}$$

Datos de salida:

Barra	Potencia Nodal Media [MW]	Demanda Media [MW]	Contribuciones de cada Generador Medio sobre Demanda Media [MW]					
1	189,66	57,91	57,91	0,00	0,00	0,00	0,00	0,00
2	248,32	182,07	7,46	0,00	82,76	0,00	0,00	91,85
3	249,01	34,66	0,00	0,00	34,66	0,00	0,00	0,00
4	117,15	117,15	53,62	0,00	30,11	0,00	0,00	33,42
5	172,15	172,15	70,67	0,00	101,49	0,00	0,00	0,00
6	125,27	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Total [MW]	-	563,94	Gen 1 [MW]	-	Gen 3 [MW]	-	-	Gen 6 [MW]
			189,66		249,01			125,27

Tabla A.4.9: Resultado de Flujos Medios sobre el Sistema de Garver de la *Figura A.4.2*. Elaboración propia.

Asignación de pérdidas a productores y consumidores				Total Pérdidas [MW]
Total Demanda Media [MW]	563,94	Total Generación Media [MW]	563,94	41,24
Total Demanda Real [MW]	532,00	Total Generación Real [MW]	573,24	
Pérdidas demandas [MW]	31,94	Pérdidas generadores [MW]	9,30	

Tabla A.4.10: Pérdidas totales del SEP de prueba según Flujos Medios. Elaboración propia.

De la *Tabla A.4.9* se observa la composición de origen que tienen los flujos demandados. Nuevamente se nota que los generadores de mayor producción son quienes abastecen a un número mayor de cargas. Al respecto, las deducciones mencionadas en el ítem anterior relativas a cargabilidad, niveles de demanda y disposición de un punto de conexión común entre producción y consumo, siguen siendo válidas. Por otra parte, la *potencia nodal media* es también consistente con el *principio de conservación de la energía*, debido a que la suma de potencias medias inyectadas es igual a la suma de potencias medias retiradas desde una misma barra.

En efecto, si se toma como ejemplo la *barra 3*, los conceptos de inyección y retiro quedan plasmados en las siguientes ecuaciones, respectivamente:

$$P_3^{media} = P_{G_3} \cong 249{,}01[MW]$$

$$P_3^{media} = P_{3-2}^{medio} + P_{3-5}^{medio} + P_{D_3}^{media} = 112{,}87 + 101{,}49 + 34{,}66 \cong 249{,}01[MW]$$

Por su parte, de la *Tabla A.4.10* se desprende que la *demanda media* tiene cuantías mayores a la demanda real debido a que adopta las pérdidas medias de la transferencia de potencia. La *generación media*, en tanto, mantiene una cuantía inferior en relación a la generación real debido a la utilización de una fracción del suministro para cubrir las pérdidas medias correspondientes. Finalmente, se aprecia que la suma de las pérdidas distribuidas tanto a generación como al parque de cargas arroja el total de las pérdidas eléctricas del sistema, haciendo que este análisis no irrumpa sobre la operación del sistema (no se han alterado los parámetros estacionarios calculados en el flujo de carga SEP). Igualmente al caso anterior, para conseguir la cantidad exacta de pérdidas destinadas a cada demanda, sólo se debe hacer la diferencia entre la cuantía *media* y real de cada barra de carga del sistema; mientras que para para cada barra de producción hay que hacer la resta entre la cuantía real y *media*.

8.4.4 Código MATLAB Ej. Sistema de Garver – M. de Bialek Flujos Medios

```
%Ejemplo Bialek 6 barras "Flujos Medios"
%-------------------------------Lectura de los datos del SEP--------------------------------
%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
hojactiva=actarch.Activesheet;
Range=get(hojactiva,'Range','D3','E8');
%Matriz de array para GD
P1=Range.value;
invoke(oparch,'Close');
invoke(actarch,'Quit');
%Matriz numérica para Gen y Dem
[M N]=size(P1);
inarch=cat(2,P1(:));
GD=reshape(inarch,M,N);
Gen=GD(:,1);
Dem=GD(:,2);

%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
hojactiva=actarch.Activesheet;
Range=get(hojactiva,'Range','G4','J10');
%Matriz de array para Red
P2=Range.value;
invoke(oparch,'Close');
invoke(actarch,'Quit');
%Matriz numérica para Red
```

```
[H I]=size(P2);
inarch=cat(2,P2(:));
Red=reshape(inarch,H,I);

Lineas=[];
Flujos=[];

for i=1:size(Red,1)
    for j=1:size(Red,2)
        if j<3
        Lineas(i,j)=Red{i,j};
        elseif j>=3
        Flujos(i,j-2)=Red{i,j};
        end
    end
end


%----------------------------Construcción matrices de entrada-----------------------------

P_G=[]; %muestra la matriz de generación del SEP "[P_G]"

for i=1:size(Gen,1)
    for j=1:size(Gen,1)
        if i==j
        P_G(i,j)=Gen{i,1};
        else
        P_G(i,j)=0;
        end
    end
end

P_D=[];%muestra la matriz de demandas de cada barra del SEP "[P_D]"

for i=1:size(Dem,1)
    for j=1:size(Dem,1)
        if i==j
        P_D(i,j)=Dem{i,1};
        else
        P_D(i,j)=0;
        end
    end
end


Pmedias_lineas=[]; % matriz de flujos medios retirados desde la barra "fila" por la barra
                     "columna".
Perdidas1=[]; %matriz de pérdidas asociadas a los flujos medios por las líneas "fila-
               columna".

for i=1:size(Gen,1)
    for j=1:size(Gen,1)

        for k=1:size(Lineas,1)
            for l=1:size(Lineas,2)

                if l==1
                    if i==Lineas(k,l)&&j==Lineas(k,l+1)
                       Pmedias_lineas(i,j)=(Flujos(k,l)+Flujos(k,l+1))/2;
                       Perdidas1(i,j)=Flujos(k,l)-Flujos(k,l+1);
                elseif l==size(Lineas,2)
                    if i==Lineas(k,l-1)&&j==Lineas(k,l)
                       Pmedias_lineas(i,j)=(Flujos(k,l-1)+Flujos(k,l))/2;
                       Perdidas1(i,j)=Flujos(k,l-1)+Flujos(k,l);
                    end
                    end
                end

            end
        end
    end
```

```
end

Pmedias_lineas=cat(2,Pmedias_lineas,zeros(6,1));
Perdidas=cat(2,Perdidas1,zeros(6,1));


Pmedias_G=zeros(size(Gen,1),size(Gen,1)); %matriz que agrupa las potencias medias de
                                            generación

for i=1:size(Gen,1) %Filas matriz [Pnodo_G]
    for j=1:size(Gen,1) %Columnas matriz [Pnodo_G]
        if i==j&&(i==1||i==3)
            Pmedias_G(i,j)=P_G(i,j)-(sum(Perdidas(i,:))/2)/2;%G1 comparte nodo con Carga1 y
                                                              G3 comparte nodo con Carga3
        elseif i==j&&i==6
            Pmedias_G(i,j)=P_G(i,j)-sum(Perdidas(i,:))/2;
        end
    end
end


Pmedias_D=zeros(size(Gen,1),size(Gen,1)); %matriz que agrupa las potencias medias de demanda

for i=1:size(Gen,1) %Filas matriz [Pnodo_D]
    for j=1:size(Gen,1) %Columnas matriz [Pnodo_D]
        if i==j&&(i==1||i==3)
       %Carga1 comparte nodo con G1 y Carga3 comparte nodo con G3
            Pmedias_D(i,j)=P_D(i,j)+(sum(Perdidas(i,:))/2)/2+(sum(Perdidas(:,j))/2);
elseif i==j&&i<6
            Pmedias_D(i,j)=P_D(i,j)+(sum(Perdidas(i,:))/2)+(sum(Perdidas(:,j))/2);
        end
    end
end


P_nodal_total=zeros(size(Gen,1),size(Gen,1)); %muestra la potencia nodal total a través de
                                                cada barra del SEP bajo Flujos Medios
for i=1:size(Gen,1)
    for j=1:size(Gen,1)
        if i==j&&(i==1||i==3||i==6)
        P_nodal_total(i,j)=Pmedias_G(i,j);
        elseif i==j&&(i==4||i==5)
        P_nodal_total(i,j)=Pmedias_D(i,j);
        elseif i==j&&i==2
        P_nodal_total(i,j)=Pmedias_D(i,j)+Pmedias_lineas(i,i+j);%Potencia absorbida por
                                                                  Carga2 más retirado 2-4
        end
    end
end




% --------------------------------Fin matrices de entrada-----------------------------------


%-----------------------------Construcción de la matriz "[Au]"-----------------------------

Au=[]; %matriz de coeficientes "[Au]", a rellenar

for i=1:size(Gen,1) %Filas matriz [Au]
    for j=1:size(Gen,1) %Columnas matriz [Au]
        if  j==i
            Au(i,j)=1;
        else
            Au(i,j)=-abs(Pmedias_lineas(j,i))/P_nodal_total(j,j);
        end
    end
end
```

```
% -------------------------Fin construcción de la matriz "[Au]"---------------------------

Au %muestra la matriz de coeficientes "[Au]"

Au_inv=Au^(-1) %muestra la inversa de la matriz de coeficientes, denotada por "[Au]^(-1)"

PG=zeros(size(Pmedias_G,1),1);
for i=1:size(Pmedias_G,1)
    PG(i,1)=Pmedias_G(i,i);
end

PG % muestra el vector de generación "[PG]"

% -------------------------------Despeje de demandas medias------------------------------

P_nodal_media=(Au^(-1))*PG

% % --------------------------Fin de despeje de demandas medias--------------------------

% % --------------------------Resultados Flujos Medios J.Bialek--------------------------

Pcarga_media=[];
for i=1:size(Pmedias_D,1)
    for j=1:size(Pmedias_D,2)
    if i==j
    Pcarga_media(i,1)=(Pmedias_D(i,j)/P_nodal_total(i,j))*(P_nodal_media(i,1));
    end
    end
end

Pcarga_media %Muestra la potencia demandada en las cargas bajo el uso de factores medios

Aportes_G=[];
for i=1:size(Au_inv,1)
    for j=1:size(Au_inv,2)
        Aportes_G(i,j)=(Pmedias_D(i,i)/P_nodal_total(i,i))*Au_inv(i,j)*PG(j,1);
    end
end

Aportes_G %Muestra los aportes de energía del Generador "columna" en la demanda media de la
               Carga "fila"


%-----------------------Registro de los resultados del Método de Bialek---------------------
%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
set(actarch,'Visible',1);%Activa la hoja de Excel
hojactiva=actarch.Activesheet;
hojaRange=get(hojactiva,'Range','D18','D23');
hojaRange1=get(hojactiva,'Range','E18','E23');
hojaRange2=get(hojactiva,'Range','F18','K23');
set(hojaRange,'Value',P_nodal_media);%Escribir Factores Medios sobre hoja Excel
set(hojaRange1,'Value',Pcarga_media);%Escribir Demandas Medias sobre hoja Excel
set(hojaRange2,'Value',Aportes_G);%Grabar los aportes de flujo de los Generadores medios
                                   sobre hoja Excel
invoke(oparch,'Close');
invoke(actarch,'Quit');
```

8.4.5 Ejemplo del Método de Bialek usando Flujos Netos

Con objeto de racionar las pérdidas eléctricas entre los generadores del sistema, J. Bialek [1] propone utilizar *flujos netos*. Para ello, se desplazan las pérdidas al extremo emisor de las líneas de transporte, con el fin de aminorar la potencia inyectada por el parque productor. El ejemplo en este caso es que la línea 1-4 podría conseguir la circulación de un *flujo neto* de 48,5$[MW]$, equivalente al flujo inyectado en la *barra 4*, bajo la premisa de entregar los 4,8$[MW]$ de pérdidas a la barra del generador G_1 (ver *Figura A.4.3*, *Tabla A.4.11* y *Tabla A.4.12*).

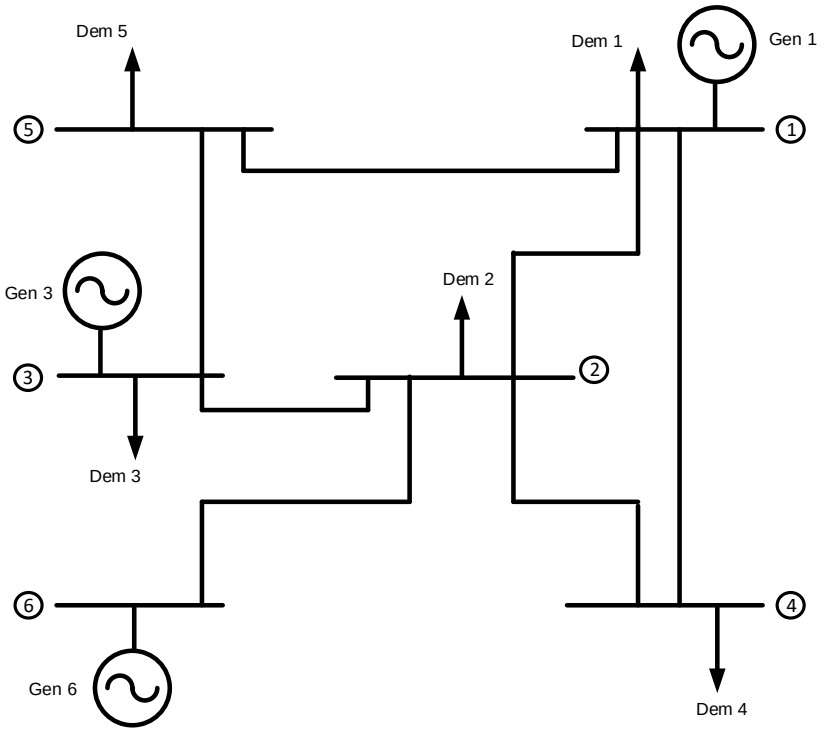


Figura A.4.3: Sistema de Garver utilizado por M. Chamba en su trabajo de título [3]. Elaboración propia.

<u>Datos de entrada:</u>

Barra	Generación [MW]	Demanda [MW]
1	191,57	56,00
2	0,00	168,00
3	255,67	28,00
4	0,00	112,00
5	0,00	168,00
6	126,00	0,00
Total [MW]	573,24	532,00
Pérdidas [MW]	41,24	

Tabla A.4.11: Datos de entrada publicados en [3].

Línea de Transmisión		Flujo de Potencia [MW]	
Barra "i"	Barra "j"	Pij_ret(i)	Pij_iny(j)
1	2	10,24	10,12
1	4	53,30	48,50
1	5	72,03	69,30
3	2	123,40	102,33
2	4	68,99	63,50
6	2	126,00	124,54
3	5	104,27	98,70
Total [MW]		558,23	516,99
Pérdidas [MW]		41,24	

Tabla A.4.12: Resultado de flujo de carga AC publicado en [3] para la revisión de métodos de tarificación.

	Balance Nodal [MW]	
Barra	Inyección	Retiro
1	191,57	191,57
2	236,99	236,99
3	255,67	255,67
4	112,00	112,00
5	168,00	168,00
6	126,00	126,00

Tabla A.4.13: Suma de retiros e inyecciones en cada barra del SEP de prueba de la *Figura A.4.3*. Elaboración propia.

Quitando todas las pérdidas del sistema de transmisión, sería factible la modelación del SEP únicamente con *flujos netos* circulando por las líneas y, con ello, podría resolverse el problema de trazabilidad mediante el *algoritmo de retiros de potencia*.

En virtud de lo anterior, es posible definir la *potencia nodal neta* en función de los *flujos netos* por las líneas y la demanda real hecha por las cargas. Para un SEP de *n* barras, se escribe:

$$P_i^{neta} = \sum_{l \in \alpha_i^{(d)}} \left| P_{i-l}^{neto} \right| + P_{\mathcal{D}_i} \quad ; \quad \forall\, i \in \{1, n\}$$

Así, en concordancia a los ejemplos mostrados en los ítems que preceden, se obtiene la siguiente igualdad en términos del tránsito de potencias retiradas desde la *barra 4*:

$$P_4^{neta} = P_{4-1}^{neto} + P_{4-2}^{neto} + P_{\mathcal{D}_4} \Leftrightarrow P_4^{neta} = 0[MW] + 0[MW] + 112[MW]$$

$$\Leftrightarrow P_4^{neta} = 112[MW]$$

Luego, dado que no existen pérdidas en el transporte, entonces $c_{li}^{neto} = \frac{|P_{l-i}^{neto}|}{P_l^{neta}} = \frac{|P_{i-l}^{neto}|}{P_l^{neta}}$. Sin embargo, dado que el *Efecto Joule* produce pérdidas ínfimas a través de las líneas, el autor [1] sugiere asumir que $\frac{|P_{l-i}^{neto}|}{P_l^{neto}} \approx \frac{|P_{l-i}|}{P_l}$, es decir, que el factor c_{ji}^{neto} podría escribirse en términos de las potencias nodales reales y de los flujos reales del SEP. De esta manera, se puede expresar que:

$$P_i^{neta} - \sum_{l \in \alpha_i^{(d)}} c_{li}^{neta} \cdot P_l^{neta} = P_{\mathcal{D}_i}$$

Dado que el despeje de la *potencia nodal neta* en la barra 4 es simple, se puede ver qué ocurre en la *barra 3* (para potencias nodales ver *Tabla A.4.13*):

$$P_3^{neta} - \left(\frac{|P_{3-2}|}{P_2} \cdot P_2^{neta} + \frac{|P_{3-5}|}{P_5} \cdot P_5^{neta} \right) = P_{\mathcal{D}_3}$$

$$\Leftrightarrow P_3^{neta} - \frac{|102{,}33|}{236{,}99} \cdot P_2^{neta} - \frac{|98{,}70|}{168{,}00} \cdot P_5^{neta} = 28{,}00$$

$$\Leftrightarrow P_3^{neta} - 0{,}4318 \cdot P_2^{neta} - 0{,}5875 \cdot P_5^{neta} = 28{,}00$$

Luego, en referencia al *algoritmo de retiros de potencia*, es posible enunciar un sistema matricial de ecuaciones que permita conocer las *potencias nodales netas* a partir de la matriz de distribución de retiros de potencia $\langle A_d \rangle$ creada a partir de los flujos reales del SEP y al vector de demanda $\langle P_{\mathcal{D}} \rangle$ que no se altera en este método, según:

$$\langle A_d \rangle \cdot \langle P_{nodal}^{neta} \rangle = \langle P_{\mathcal{D}} \rangle$$

De este modo, conforme a lo expresa J. Bialek en su documento [1], para el tránsito de potencia por la red de transmisión se puede establecer la relación:

$$\left| P_{i-l}^{neto} \right| = \frac{\left| P_{i-l}^{neto} \right|}{P_i^{neta}} \cdot P_i^{neta} \cong \frac{|P_{i-l}|}{P_i} \cdot P_i^{neta} = \frac{|P_{i-l}|}{P_i} \cdot \sum_{k=1}^{n} A_{d_{ik}}^{-1} \cdot P_{\mathcal{D}_k} \;\; ; \quad \forall\, j \in \alpha_i^{(u)}$$

Mientras que para los generadores la igualdad viene definida por:

$$P_{G_i}^{neta} = \frac{P_{G_i}^{neta}}{P_i^{neta}} \cdot P_i^{neta} \cong \frac{P_{G_i}}{P_i} \cdot P_i^{neta} = \frac{P_{G_i}}{P_i} \cdot \sum_{k=1}^{n} A_{d_{ik}}^{-1} \cdot P_{\mathcal{D}_k} \quad ; \quad \forall\, i \in \{1, n\}$$

Posteriormente, las pérdidas eléctricas repartidas dentro del bloque de generación se podrán despejar de acuerdo con:

$$G:\ \Delta P_{loss_i} = P_{G_i} - P_{G_i}^{neta}$$

De forma análoga a los procedimientos detallados en los ítems 8.4.1 y 8.4.3, la cuantía de las *potencias nodales netas* no siempre se puede conseguir por mera inspección visual, por lo que se ha implementado un código en MATLAB para dar con los resultados del SEP de la *Figura A.4.3* (ver ítem 8.4.6), cuyo archivo ".m" ha sido entregado como material complementario a este documento.

El detalle del sistema matricial y los datos de salida se detallan a continuación:

<u>Formulación del problema:</u>

$$\langle A_d \rangle \cdot \langle P_{nodal}^{neta} \rangle = \langle P_{\mathcal{D}} \rangle$$

$$\begin{pmatrix} 1 & -0{,}0427 & 0 & -0{,}4330 & -0{,}4125 & 0 \\ 0 & 1 & 0 & -0{,}5670 & 0 & 0 \\ 0 & -0{,}4318 & 1 & 0 & -0{,}5875 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -0{,}5255 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} P_1^{neta} \\ P_2^{neta} \\ P_3^{neta} \\ P_4^{neta} \\ P_5^{neta} \\ P_6^{neta} \end{pmatrix} = \begin{pmatrix} 56{,}00 \\ 168{,}00 \\ 28{,}00 \\ 112{,}00 \\ 168{,}00 \\ 0 \end{pmatrix}$$

<u>Datos de salida:</u>

Barra	Potencia Nodal Neta [MW]	Generación Neta [MW]	Contribuciones de cada Carga Neta sobre Generación Neta [MW]					
1	183,69	183,69	56,00	7,17	0,00	51,21	69,30	0,00
2	231,50	0,00	0,00	0,00	0,00	0,00	0,00	0,00
3	226,66	226,66	0,00	72,54	28,00	27,42	98,70	0,00
4	112,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
5	168,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
6	121,65	121,65	0,00	88,29	0,00	33,37	0,00	0,00
Total [MW]	-	532,00	Dem 1 [MW]	Dem 2 [MW]	Dem 3 [MW]	Dem 4 [MW]	Dem 5 [MW]	-
			56,00	168,00	28,00	112,00	168,00	

Tabla A.4.14: Resultado de Flujos Netos sobre el Sistema de Garver de la *Figura A.4.3*. Elaboración propia.

Asignación de pérdidas a productores	
Total Generación Neta [MW]	532,00
Total Generación Real [MW]	573,24
Pérdidas [MW]	41,24

Tabla A.4.15: Total de pérdidas del Sistema de Garver de la *Figura A.4.3*. Elaboración propia.

De la *Tabla A.4.14* se observa que la *potencia nodal neta* es consistente al supuesto planteado en el Método de Bialek vía Flujos Netos, por cuanto el balance de energía nodal igualmente cumple con el *principio de conservación de la energía*.

En otro tema, la *Tabla A.4.14* también expone los requerimientos de potencia que hace el parque de demandas sobre los flujos suministrados por la generación. En ese sentido, se nota que las exigencias de potencia permiten el abastecimiento pleno o mayoritario de la demanda cuando exista una conexión directa (no necesariamente inmediata) entre la producción y el consumo, así como también de comportamientos propios de la transmisión (capacidad de transferencia de potencia) para garantizar una mayor participación de un *generador neto* en la cobertura de una demanda particular. Lo anterior lo ejemplifica bien $\mathcal{D}_2$, quien se abastece de manera importante por G_3 y G_6 con quienes tiene una conexión directa, no obstante, y a pesar de que $\mathcal{D}_2$ también está acoplado a la barra 1, se observa que G_1 atiende tan sólo una pequeña fracción de la demanda de la carga 2.

Además, es importante resaltar que en este caso se pueden extraer conclusiones similares a las realizadas con el ejemplo de *flujos gruesos* (ítem 8.4.1). Por ejemplo, en la *Tabla A.4.15* se exhiben las pérdidas del SEP de prueba bajo la metodología de *flujos netos*. Estas confirman ser iguales a las pérdidas eléctricas extraídas desde el flujo de carga en estado permanente del "Modelo de Garver", además de exhibir la disminución del suministro energético debido a que el *parque generador neto* debe hacerse cargo de los costos físicos cometidos durante la etapa de transporte. Para obtener la cantidad exacta de pérdidas destinadas a cada planta de producción, sólo se debe hacer la diferencia entre la cuantía real y *neta* de cada barra de generación del sistema.

8.4.6 Código MATLAB Ej. Sistema de Garver – M. de Bialek Flujos Netos

```
%Ejemplo Bialek 6 barras "Flujos Netos"
%---------------------------------Lectura de los datos del SEP-----------------------------
%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
hojactiva=actarch.Activesheet;
Range=get(hojactiva,'Range','D3','E8');
%Matriz de array para GD
```

```
P1=Range.value;
invoke(oparch,'Close');
invoke(actarch,'Quit');
%Matriz numérica para Gen y Dem
[M N]=size(P1);
inarch=cat(2,P1(:));
GD=reshape(inarch,M,N);
Gen=GD(:,1);
Dem=GD(:,2);

%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
hojactiva=actarch.Activesheet;
Range=get(hojactiva,'Range','G4','J10');
%Matriz de array para Red
P2=Range.value;
invoke(oparch,'Close');
invoke(actarch,'Quit');
%Matriz numérica para Red
[H I]=size(P2);
inarch=cat(2,P2(:));
Red=reshape(inarch,H,I);

Lineas=[];
Flujos=[];

for i=1:size(Red,1)
    for j=1:size(Red,2)
        if j<3
        Lineas(i,j)=Red{i,j};
        elseif j>=3
        Flujos(i,j-2)=Red{i,j};
        end
    end
end


Nodoret=Lineas(:,1);
Nodoiny=Lineas(:,2);

%------------------------------Construcción matrices de entrada-----------------------------

P_G=[]; %muestra la matriz de generación del SEP "[P_G]"

for i=1:size(Gen,1)
    for j=1:size(Gen,1)
        if i==j
        P_G(i,j)=Gen{i,1};
        else
        P_G(i,j)=0;
        end
    end
end


P_D=[];%muestra la matriz de demandas de cada barra del SEP "[P_D]"

for i=1:size(Dem,1)
    for j=1:size(Dem,1)
        if i==j
        P_D(i,j)=Dem{i,1};
        else
        P_D(i,j)=0;
        end
    end
end
```

```
P_ret_barra1=[]; % matriz de flujos retirados desde la barra "fila" por la barra "columna".

for i=1:size(Gen,1)
    for j=1:size(Gen,1)

        for k=1:size(Lineas,1)
            for l=1:size(Lineas,2)

                if l==1
                    if i==Lineas(k,l)&&j==Lineas(k,l+1)

                        if Flujos(k,l)>Flujos(k,l+1)
                        P_ret_barra1(i,j)=Flujos(k,l);
                       elseif Flujos(k,l+1)>Flujos(k,l)
                        P_ret_barra1(i,j)=Flujos(k,l+1);
                        end
                elseif l==size(Lineas,2)
                    if i==Lineas(k,l-1)&&j==Lineas(k,l)

                        if Flujos(k,l-1)>Flujos(k,l)
                        P_ret_barra1(i,j)=Flujos(k,l-1);
                       elseif Flujos(k,l)>Flujos(k,l-1)
                        P_ret_barra1(i,j)=Flujos(k,l);
                        end
                    end
                    end
                end
            end
        end
    end
end

P_ret_barra=cat(2,P_ret_barra1,zeros(6,1));

P_iny_barra1=[]; %matriz de flujos inyectados en la barra "columna" por la barra "fila".

for i=1:size(Gen,1)
    for j=1:size(Gen,1)

        for k=1:size(Lineas,1)
            for l=1:size(Lineas,2)

                if l==1
                    if i==Lineas(k,l)&&j==Lineas(k,l+1)

                        if Flujos(k,l)<Flujos(k,l+1)
                        P_iny_barra1(i,j)=Flujos(k,l);
                       elseif Flujos(k,l+1)<Flujos(k,l)
                        P_iny_barra1(i,j)=Flujos(k,l+1);
                        end

                elseif l==size(Lineas,2)
                    if i==Lineas(k,l-1)&&j==Lineas(k,l)

                        if Flujos(k,l-1)<Flujos(k,l)
                        P_iny_barra1(i,j)=Flujos(k,l-1);
                       elseif Flujos(k,l)<Flujos(k,l-1)
                        P_iny_barra1(i,j)=Flujos(k,l);
                        end

                    end
                    end
                end
            end
        end
    end
end


P_iny_barra=cat(2,P_iny_barra1,zeros(6,1));
```

```
Pnodo_G=zeros(size(Gen,1),size(Gen,1)); %matriz que agrupa las potencias totales a través de
                                          los nodos de generación

for i=1:size(Gen,1) %Filas matriz [Pnodo_G]
    for j=1:size(Gen,1) %Columnas matriz [Pnodo_G]
        if i==j&&i==1
            Pnodo_G(i,j)=P_G(i,j)+sum(P_iny_barra(:,j));
        elseif i==j&&i==3
            Pnodo_G(i,j)=P_G(i,j)+sum(P_iny_barra(:,j));
        elseif i==j&&i==6
            Pnodo_G(i,j)=P_G(i,j)+sum(P_iny_barra(:,j));
        end
    end
end


Pnodo_D=zeros(size(Gen,1),size(Gen,1)); %matriz que agrupa las potencias totales a través de
                                          los nodos de demanda
for i=1:size(Gen,1) %Filas matriz [Pnodo_D]
    for j=1:size(Gen,1) %Columnas matriz [Pnodo_D]
        if i==j&&i==2
            Pnodo_D(i,j)=P_D(i,j)+sum(P_ret_barra(i,:));
        elseif i==j&&i==4
            Pnodo_D(i,j)=P_D(i,j)+sum(P_ret_barra(i,:));
        elseif i==j&&i==5
            Pnodo_D(i,j)=P_D(i,j)+sum(P_ret_barra(i,:));
        end
    end
end

P_nodal_total=Pnodo_G+Pnodo_D; %muestra la potencia nodal neta a través de cada barra del
                                    SEP bajo Flujos Gruesos


% -------------------------------Fin matrices de entrada-----------------------------------


%----------------------------Construcción de la matriz "[Ad]"-----------------------------

Ad=[]; %matriz de coeficientes "[Ad]", a rellenar

for i=1:size(Gen,1) %Filas matriz [Ad]
    for j=1:size(Gen,1) %Columnas matriz [Ad]
        if  j==i
            Ad(i,j)=1;
        else
            Ad(i,j)=-abs(P_iny_barra(i,j))/P_nodal_total(j,j);
        end
    end
end


% -------------------------Fin construcción de la matriz "[Ad]"----------------------------

Ad %muestra la matriz de coeficientes "[Ad]"

Ad_inv=Ad^(-1) %muestra la inversa de la matriz de coeficientes, denotada por "[Ad]^(-1)"

PD=zeros(size(P_D,1),1);
for i=1:size(P_D,1)
    PD(i,1)=P_D(i,i);
end

PD % muestra el vector de demanda "[PD]"


% -------------------------------Despeje de generadores netos-----------------------------

P_nodal_neta=(Ad^(-1))*PD

% % -------------------------Fin de despeje de generación neta----------------------------
```

```
% % --------------------------Resultados Flujos Netos J.Bialek--------------------------

Pgeneracion_neta=[];
for i=1:size(Gen,1)
    Pgeneracion_neta(i,1)=(Gen{i,1}/P_nodal_total(i,i))*(P_nodal_neta(i,1));
end

Pgeneracion_neta %Muestra la potencia suministrada por los generadores bajo el uso de
                      factores netos


Aportes_D=[];
for i=1:size(Ad_inv,1)
    for j=1:size(Ad_inv,2)
        Aportes_D(i,j)=(Gen{i,1}/P_nodal_total(i,i))*Ad_inv(i,j)*PD(j,1);
    end
end

Aportes_D %Muestra los requerimientos de energía que la carga "columna" exige al generador
               neto "fila"


%--------------------------Registro de los resultados del Método de Bialek-----------------
%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
set(actarch,'Visible',1);%Activa la hoja de Excel
hojactiva=actarch.Activesheet;
hojaRange=get(hojactiva,'Range','D18','D23');
hojaRange1=get(hojactiva,'Range','E18','E23');
hojaRange2=get(hojactiva,'Range','F18','K23');
set(hojaRange,'Value',P_nodal_neta);%Escribir Factores Netos sobre hoja Excel
set(hojaRange1,'Value',Pgeneracion_neta);%Escribir Generadores Netos sobre hoja Excel
set(hojaRange2,'Value',Aportes_D);%Grabar exigencias de flujo de las Cargas sobre hoja Excel
invoke(oparch,'Close');
invoke(actarch,'Quit');
```

8.4.7 Método de Kirschen, Allan & Strbac

En febrero de 1997 D. Kirschen, R. Allan y G. Strbac publicaron su trabajo titulado *Contributions of individual generators to loads and flows* [2], en donde se describe una metodología para conocer el origen de las inyecciones de potencia que conforman el flujo demandado por una carga particular y, con ello, aportar información relevante para la trazabilidad de electricidad. También, el procedimiento permite estar al tanto del nivel de participación que cada fuente tiene en la producción de pérdidas eléctricas asociadas a la utilización del sistema de transmisión, con lo cual el costo físico se atribuye únicamente a los generadores [4]. Adicionalmente, de acuerdo con lo que sostienen los autores [2], este método se puede aplicar para analizar potencias activas o reactivas y su provecho está abierto incluso para métodos incrementales.

El procedimiento se basa en la agrupación de aquellas barras que se abastecen de la potencia inyectada de la misma fuente, para lo cual se debe contar previamente con los resultados de flujo de carga en

estado permanente del sistema. Posteriormente, basándose en el *principio de repartición proporcional*, es factible conocer la composición de los flujos de potencia de las cargas y líneas de la red en función de las contribuciones de suministro procedente del parque generador del SEP.

En términos concretos, el algoritmo hace uso de tres conceptos definidos por D. Kirschen, R. Allan y G. Strbac [2] para modelar el rastreo de electricidad en la red, a saber: *dominio de un generador*, *área común de generación* y *enlaces del sistema*. Con el fin de explicar la metodología, en los siguientes puntos se enseñan las definiciones de estos conceptos y, en la sección Anexos de este trabajo, se expone la aplicación de los mismos en una adaptación del ejemplo planteado por los autores.

8.4.7.1 Conceptos fundamentales

8.4.7.1.1 Dominio de un generador

La publicación [2] define el *dominio de un generador* como el conjunto de barras que son abastecidas según el *alcance* de un generador puntual. El *alcance*, según los autores, tiene cabida si es que existe un camino físico a través de la red que permita "unir" el punto de suministro con el punto de consumo de potencia. Para conocer el *dominio de un generador*, se recomienda usar un algoritmo que enlista todas las barras de la red vinculadas a un generador puntual para luego someterlas a una evaluación.

El arreglo de instrucciones es como sigue:

```
Inicio del algoritmo:
Escoja una barra que esté conectada a un generador y colóquela en una lista

        Inicio de condición 1:
        Mientras haya barras en la lista, haga lo siguiente:
                {Remueva la primera barra de la lista y agréguela al dominio del
        generador}

                    Inicio de ciclo:
                       Repita el siguiente paso para todas las líneas conectadas a dicha
                    barra:
                            {
                            Inicio condición 2:
                            Si la potencia por esa línea sale de la barra y si la barra al
                            otro extremo de la línea aún no ha sido ingresada en el dominio
                            del generador, entonces:
                            {Agregue la barra opuesta a la lista}
                            Fin condición 2
                            }
                    Fin de ciclo

        Fin condición 1

Fin del algoritmo
```

Aquí es bueno enfatizar en que el *dominio de un generador* para rastrear flujos activos no coincide, necesariamente, con el *dominio* utilizado para trazar flujos reactivos.

8.4.7.1.2 Área común de generación

Si bien el concepto de *dominio de un generador* es un concepto interesante, por sí solo es un concepto insuficiente para responder al problema de trazabilidad, debido a la eventual superposición de los diferentes *dominios* del sistema. En ese sentido, con el afán de incorporar elementos que permitan discriminar la influencia de cada generador en los flujos de potencia del SEP, se define un nuevo concepto: el *área común de generación*.

Según se explica en *Contributions of individual generators to loads and flows* [2], el *área común de generación* corresponde al conjunto de barras alimentadas directamente por uno o más generadores, vale decir, equivale al conjunto de todas aquellas barras que tienen contacto directo, vía línea de transmisión, con el nodo de generación. Dentro de la definición se especifica que aquellas barras que no están conectadas entre si y que sin embargo son alimentadas por un mismo generador o grupo de ellos, son vinculadas a *áreas comunes* distintas; forzando a que cada barra del sistema pertenezca a una única *área común de generación*. Adicionalmente, a la cardinalidad de cada uno de estos conjuntos se le denomina *rango* y corresponde a un número natural acotado entre 1 y la máxima cantidad de generadores que conviven en el *área común*.

El arreglo de instrucciones que permiten conocer el *área común* de los diversos generadores de un SEP se muestra a continuación:

```
Inicio del algoritmo:
Determine el dominio de cada generador del sistema. Luego, para cada nodo del SEP,
registre los generadores que permiten que este se abastezca

        Inicio de ciclo:
        Repita la siguiente instrucción para cada barra de la red:

                Inicio de condición:
                {
                Si la barra aún no pertenece a un área común, entonces:
                Invente una nueva área común basada en el o los generadores que alimentan
                dicho nodo
                Propague, de manera recursiva, el área común recientemente concebida a
                todos los nodos conectados a la barra en estudio
                Fin condición
                }
        Fin de ciclo

Fin del algoritmo
```

8.4.7.1.3 Enlaces del sistema

Finalmente se define el concepto de *enlace del sistema*, que se define como un elemento para conectar las distintas barras de la red. Un *enlace* puede permitir la conexión entre dos nodos pertenecientes a una misma *área común de generación*, así como también favorecer la conexión entre dos barras de *áreas comunes* distintas.

Conforme a lo que se explica en el trabajo [2], por un *enlace* pueden fluir potencias de diversas líneas bajo dos condiciones: que el tránsito de dichas potencias tenga igual sentido y que este se efectúe desde un *área común* de menor *rango* a otra de mayor *rango*. Esta última condición implica que el tránsito de potencia se efectúa desde una zona que posee una cantidad menor de generadores hacia otra de cardinalidad mayor.

8.4.7.2 Representación gráfica del sistema

Para estudiar el nivel de uso que tienen los agentes del SEP en cada transferencia energética y definir la composición de origen que tienen los flujos demandados, D. Kirschen, *et al*. [2] proponen la representación del SEP mediante un esquema que logra compactar las *áreas comunes de generación* y los *enlaces del sistema* en un gráfico de puntos y ramas, respectivamente.

El resultado es un gráfico directo y acíclico denominado *gráfico de estado del sistema*, el cual ilustra las relaciones de potencia en régimen permanente que surgen luego de aplicar un flujo de carga a la red en ausencia de pérdidas. Se dice que la representación gráfica es "directa" porque el sentido en el que fluye la potencia por los *enlaces* es propio de la configuración del sistema. Mientras que se afirma que esta también es "acíclica" debido a que los *enlaces* pueden conectar *áreas comunes* bajo la condición de *rangos* descrita en el punto inmediatamente anterior (ítem 4.2.1.3).

Adicionalmente es bueno comentar que el *gráfico de estado del sistema* comúnmente viene definido por un punto inicial o "nodo raíz" correspondiente al *área común de generación* que posee *rango* igual a 1 y que coincide, por tanto, con el generador que está ubicado más "aguas arriba" en la red de transmisión.

8.4.7.3 Método recursivo del sistema

Con el fin de aportar información cuantitativa al modelo, se hace uso de nuevas definiciones que terminan por constituir un algoritmo recursivo. En ese sentido, se define el *flujo interno de un área común* como la sumatoria de las potencias inyectadas por los generadores dentro de dicha zona, más las potencias importadas desde otras *áreas comunes* a través de los *enlaces del sistema*; arrojando siempre un número real positivo. Adicionalmente, se define el *flujo externo de un área común* como la sumatoria

de las potencias exportadas desde dicha zona, mediante los *enlaces del sistema*, hacia *áreas comunes* de mayor *rango*.

Luego, con las definiciones recientemente enunciadas y en coherencia al *principio de conservación de la energía*, el "paper" [2] explicita el siguiente supuesto:

<< Para un *área común* dada, si la proporción de *flujo interno* que se puede rastrear desde el generador G_i es x_i , entonces la proporción de *flujo externo* que se puede trazar desde el generador G_i también es x_i >>

Supuesto que puede reformularse para enunciar el siguiente corolario:

<< Para un *área común* dada, si la proporción de *flujo interno* que se puede rastrear desde G_i es x_i , entonces la proporción del flujo demandado por una carga vinculada al generador G_i también es x_i >>

Aquí es bueno comentar que las suposiciones antes mencionadas no pueden ser probadas ni desaprobadas *a priori* y que la justificación de esto es que el *supuesto de repartición proporcional* parece ser más razonable que cualquier otra suposición posible [2].

De las ideas recientemente presentadas, se extrae que el *flujo interno de un área común* es igual a la suma de los *flujos externos* más la demanda de las cargas de dicha zona, lo que tiene un impacto importante en la formulación de las relaciones cuantitativas. Efectivamente, con los supuestos es posible crear un *método recursivo* para conocer las cantidades de potencia que cada generador puntual aporta a un *área común* particular, como se expone a continuación:

$$\mathcal{f}_{G_i jk} = c_{G_i j} \cdot \mathcal{f}_{jk}$$

$$\mathcal{I}_k = \sum_j \mathcal{f}_{jk}$$

$$c_{G_i k} = \frac{\sum_j \mathcal{f}_{G_i jk}}{\mathcal{I}_k}$$

Donde:

$c_{G_i j}$: contribución del generador G_i al *flujo interno* del *área común j*.

$c_{G_i k}$: contribución del generador G_i a la carga y *flujo externo* del *área común k*.

$\mathcal{f}_{jk}$: flujo desde el *área común j* al *k* a través de un *enlace*.

$\mathcal{f}_{G_i jk}$: flujo desde el *área común j* al *k* por medio del *enlace* proveniente del generador G_i.

$\mathcal{I}_k$: *flujo interno* del *área común k*

Finalmente, se debe comentar que para iniciar el cálculo recursivo se utilizan los "nodos raíces" (en donde se conoce el *flujo interno* a cabalidad) sobre los que luego se calcula el *flujo externo* asociado. Posteriormente, el *método recursivo* contempla la propagación del cálculo hacia las *áreas comunes* de mayor *rango*. Con el fin de ejemplificar el uso de esta metodología, en el ítem 8.4.8 se usa un SEP de 6 barras sobre el cual se aplican los conceptos presentados a lo largo de este ítem. El sistema utilizado se ha extraído desde el trabajo de los autores [2], pero las inyecciones y retiros se han definido arbitrariamente. Con el desarrollo del problema se persigue mostrar cómo se compone el flujo demandado según el *área común* y cómo se reparte la totalidad de pérdidas entre los generadores. Adicionalmente, en el ítem 8.4.9 se expone el código empleado en MATLAB para la ejecución del Método de Kirschen, Allan & Strbac sobre el sistema de prueba.

8.4.8 Ejemplo del Método de Kirschen, Allan y Strbac

Para mostrar la aplicación del método, los autores [2] proponen considerar un SEP de 6 barras y 9 líneas cuya adaptación, propuesta en este trabajo, se expone en la *Figura A.4.4*. Los datos de entrada, obtenidos desde un flujo de carga AC vía software DigSilent PF (archivo entregado como material complementario a este trabajo), se exhiben en la *Tabla A.4.16* y *Tabla A.4.17* para la producción, consumo y flujos, respectivamente.

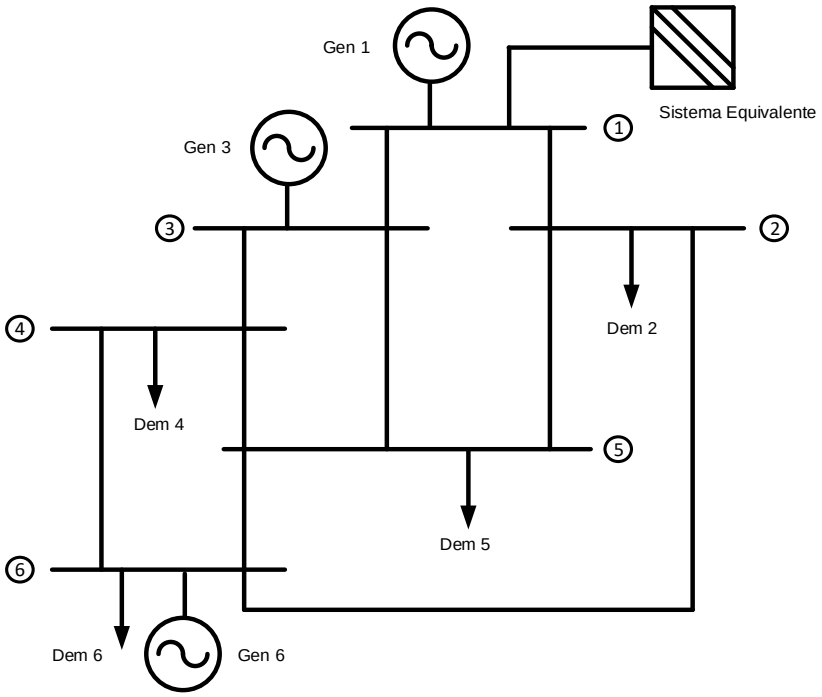


Figura A.4.4: Adaptación del SEP de prueba utilizado en [2] para aplicar la metodología propuesta. Elaboración propia.

Barra	Generación [MW]	Demanda [MW]
1	60,00	-1,09
2	0,00	20,00
3	50,00	0,00
4	0,00	10,00
5	0,00	20,00
6	10,00	70,00
Total [MW]	120,00	118,91
Pérdidas [MW]	1,09	

Tabla A.4.16: Datos de entrada del sistema de la *Figura A.4.4.* Elaboración propia.

Línea de Transmisión		Flujo de Potencia [MW]	
Barra "i"	Barra "j"	Pij_ret(i)	Pij_iny(j)
1	2	48,35	48,11
1	3	12,74	12,68
2	5	23,25	23,12
2	6	4,86	4,81
3	4	49,47	49,37
3	5	13,21	13,13
4	5	18,14	18,06
4	6	21,22	21,04
5	6	34,30	34,15
Total [MW]		225,55	224,46
Pérdidas [MW]		1,09	

Tabla A.4.17: Flujo de carga AC obtenido desde DigSilent PF. Elaboración propia.

Nótese que la información antes enseñada contempla la conexión de un "Sistema Equivalente" con el fin de asumir las pérdidas del sistema sin que las cantidades de generación y/o demanda propuestas por los autores [2] experimenten cambios en la resolución del flujo de carga a través del software.

Con lo anterior, el uso de los conceptos arroja los siguientes resultados:

<u>Dominio de los generadores:</u>

Dominio G_1: sistema completo; barras 1, 2, 3, 4, 5 y 6

Dominio G_3: barras 3, 4, 5 y 6

Dominio G_6: barra 6

Áreas comunes de generación y rangos:

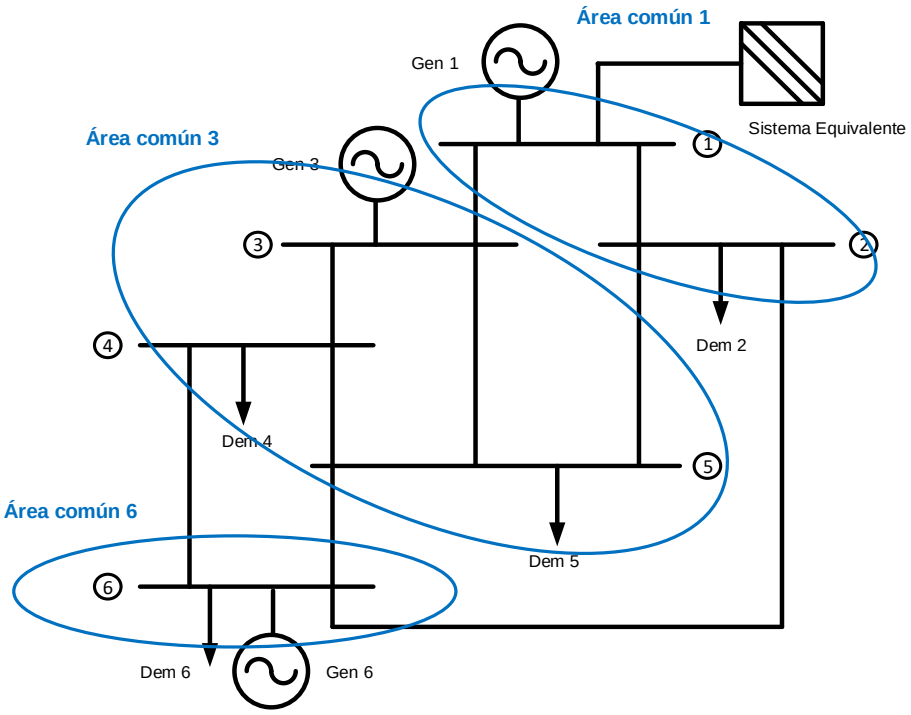


Figura A.4.5: Determinación de *áreas comunes de generación* en el SEP de la *Figura A.4.4*. Elaboración propia.

En base a la imagen que precede, se concluye que:

Área común 1: barras 1 y 2 alimentadas por G_1. *Rango:* 1

Área común 3: barras 3, 4 y 5 alimentadas por G_1 y G_3. *Rango:* 2

Área común 6: barra 6 alimentada por G_1 , G_3 y G_6. *Rango:* 3

Enlaces del sistema:

Enlace 1-3 (conecta las *áreas comunes 1* y *3*): línea 1-3 y línea 2-5, igual a $35{,}987[MW]$

Enlace 3-6 (conecta las *áreas comunes 3* y *6*): línea 4-6 y línea 5-6, igual a $55{,}52[MW]$

Enlace 1-6 (conecta las *áreas comunes 1* y *6*): línea 2-6, igual a $4{,}86[MW]$

Se observa además que la línea de transmisión 1-2 es interna al *área común 1*; las líneas 3-4, 3-5 y 4-5 pertenecen al *área común 3*; mientras que el *área común 6* no posee líneas internas, ya que los tramos 2-6, 4-6 y 5-6 atraviesan *áreas comunes* distintas.

Gráfico de estado del sistema:

Si se contempla que, posterior a la aplicación de los conceptos, el SEP de prueba queda como se exhibe en la *Figura A.4.6*, entonces el gráfico directo y acíclico para el estado del sistema tendría la forma y las cantidades que se presentan en el siguiente esquema:

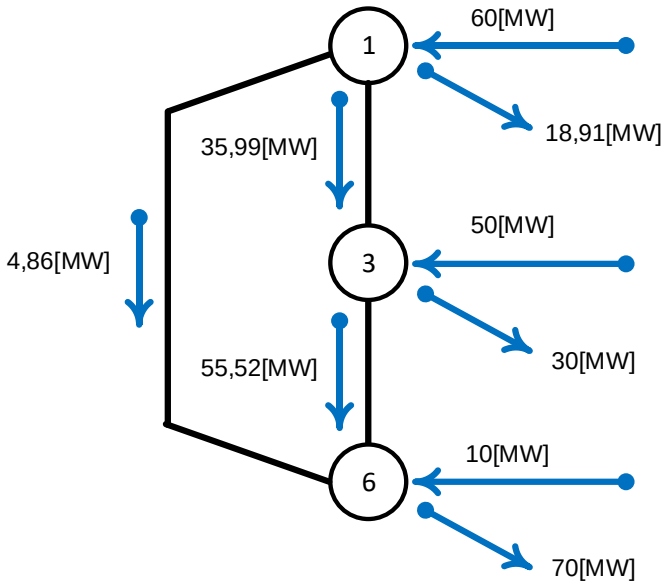


Figura A.4.6: Gráfico de puntos y ramas. Relaciones de potencia en ausencia de pérdidas. Elaboración propia.

La *Figura A.4.7* expone las *áreas comunes de generación* encerradas en círculos (puntos), hacia los cuales entra la potencia total inyectada y sale la potencia total retirada. También se puede ver la potencia que transita por cada *enlace*, correspondiente a la suma de los flujos implicados entre cada *área común*. En relación con ello, debe notarse que el *principio de repartición proporcional* se cumple a cabalidad, ya que el balance energético en cada punto permite reconocer las pérdidas parciales incurridas en cada *área común de generación* y, con ello, asumir las pérdidas totales del sistema (cuantía afín al valor enunciado en la *Tabla A.4.16* y *Tabla A.4.17*, respectivamente).

En efecto:

Área común 1: $60 = 4{,}86 + 35{,}99 + 18{,}91 + P_{loss_1} \Leftrightarrow P_{loss_1} \cong 0{,}24[MW]$

Área común 3: $50 + 35{,}99 = 55{,}52 + 30 + P_{loss_3} \Leftrightarrow P_{loss_3} \cong 0{,}47[MW]$

Área común 6: $4{,}86 + 55{,}52 + 10 = 70 + P_{loss_6} \Leftrightarrow P_{loss_6} \cong 0{,}38[MW]$

Pérdidas totales del SEP: $P_{loss_{total}} = P_{loss_1} + P_{loss_3} + P_{loss_6} \Leftrightarrow P_{loss_{total}} \cong 1{,}09[MW]$

Adicionalmente, y en sintonía con lo comentado por F. Danitz en [4], queda en evidencia que ante cualquier cambio en el sistema estudiado (a saber, fallas, contingencias, cambio de *taps* en transformadores y todo evento que termine por modificar el flujo de potencia por una línea del SEP), la cuantía de los flujos de potencia que transitan por los *enlaces del sistema* podría sufrir cambios importantes, debido a que los *dominios*, *flujos internos*, *flujos externos* y *áreas comunes* del modelo se verían alterados. Lo anterior culminaría por cambiar drásticamente el *gráfico de estado del sistema*.

<u>Método recursivo del sistema:</u>

Del *gráfico de estado del sistema* que se acaba de exponer, es posible extraer las siguientes relaciones:

i) *Flujos internos* de cada *área común*:

En virtud del *principio de conservación de la energía* (sin pérdidas), es posible plantear que las inyecciones de potencia a los nodos del esquema son:

- *Área común 1*: $\mathcal{I}_1 = \mathcal{f}_{G_A 11} = 60[MW]$
- *Área común 3*: $\mathcal{I}_3 = \mathcal{f}_{G_B 33} + \mathcal{f}_{13} = 50[MW] + 35{,}99[MW] = 85{,}99[MW]$
- *Área común 6*: $\mathcal{I}_6 = \mathcal{f}_{G_C 66} + \mathcal{f}_{16} + \mathcal{f}_{36} = 10[MW] + 4{,}86[MW] + 55{,}52[MW]$
  $\Leftrightarrow \mathcal{I}_6 = 70{,}38[MW]$

ii) Cálculo de contribuciones de cada generador:

Basándose en las relaciones expuestas en el ítem 4.2.3, se logran los siguientes resultados:

- Contribución relativa sobre las cargas y el *flujo externo* del *área común 1* (nodo raíz):

$$G_1: \ \mathcal{f}_{G_1 11} = 60[MW]$$

$$G_1: \ c_{G_1 1} = \frac{\mathcal{f}_{G_1 11}}{\mathcal{I}_1} = \frac{60}{60} = 1\ pu$$

- Contribución absoluta al *flujo interno* del *área común 3*:

$$G_1: \ \mathcal{f}_{G_1 13} = c_{G_1 1} \cdot \mathcal{f}_{13} = 1 \cdot 35{,}99 = 35{,}99[MW]$$

$$G_3: \ \mathcal{f}_{G_3 33} = 50[MW]$$

- Contribución relativa sobre las cargas y el *flujo externo* del *área común 3*:

$$G_1:\ \ c_{G_1 3} = \frac{\mathcal{P}_{G_1 13}}{\mathcal{J}_3} = \frac{35{,}99}{85{,}99} \cong 0{,}4185\ pu$$

$$G_3:\ \ c_{G_3 3} = \frac{\mathcal{P}_{G_3 33}}{\mathcal{J}_3} = \frac{50}{85{,}99} \cong 0{,}5815\ pu$$

- Contribución absoluta al *flujo interno* del *área común 6*:

$$G_1:\ \ \mathcal{P}_{G_1 16} = c_{G_1 1} \cdot \mathcal{P}_{16} + c_{G_1 3} \cdot \mathcal{P}_{36} = 1 \cdot 4{,}86\ +\ 0{,}4185 \cdot 55{,}52\ \cong 28{,}10[MW]$$

$$G_3:\ \ \mathcal{P}_{G_3 36} = c_{G_3 3} \cdot \mathcal{P}_{36} = 0{,}5815 \cdot 55{,}52\ \cong 32{,}29[MW]$$

$$G_6:\ \ \mathcal{P}_{G_6 66} = 10[MW]$$

- Contribución relativa sobre las cargas y el *flujo externo* del *área común 6*:

$$G_1:\ \ c_{G_1 6} = \frac{\mathcal{P}_{G_1 16}}{\mathcal{J}_6} = \frac{28{,}10}{70{,}38} \cong 0{,}3993\ pu$$

$$G_3:\ \ c_{G_3 6} = \frac{\mathcal{P}_{G_3 36}}{\mathcal{J}_6} = \frac{32{,}29}{70{,}38} \cong 0{,}4588\ pu$$

$$G_6:\ \ c_{G_6 6} = \frac{\mathcal{P}_{G_6 66}}{\mathcal{J}_6} = \frac{10}{70{,}38} \cong 0{,}1421\ pu$$

Hasta ahora, las cantidades en pu enunciadas con anterioridad han permitido conocer los aportes absolutos (en $[MW]$) que cada generador hace sobre los *flujos externos* que ingresan a determinada *área común de generación*. Del mismo modo, en virtud del *principio de repartición proporcional* y de las cantidades pu arrojadas por el método, es posible determinar las contribuciones de potencia del parque generador sobre los flujos que circulan por las líneas y cargas del sistema.

En ese sentido, si bien el SEP que se ilustra en la *Figura A.4.4* es reducido y responde a un sistema de prueba, se ha implementado un código base en MATLAB para la obtención de los resultados (ver ítem 8.4.8) y el archivo ".m" ha sido entregado como material complementario a este trabajo.

Las cantidades más relevantes se presentan en los cuadros *Tabla A.4.18*, *Tabla A.4.19* y *Tabla A.4.20*.

Contribuciones relativas de cada Generador sobre las cargas y flujos externos de cada "Á.C. de G." en p.u.			
Área Común de Generación			
Generador	1	3	6
Gen 1	1,0000	0,4185	0,3992
Gen 3	0,0000	0,5815	0,4587
Gen 6	0,0000	0,0000	0,1421
Total [pu]	1,0000	1,0000	1,0000

Tabla A.4.18: Resultado en "*pu*" del Método de Kirschen, Allan y Strbac sobre el SEP de prueba. Elaboración propia.

Contribuciones absolutas de cada Generador sobre la Demanda de cada "Á.C. de G." en [MW]			
Área Común de Generación			
Generador	1	3	6
Gen 1	18,91	12,56	27,94
Gen 3	0,00	17,44	32,11
Gen 6	0,00	0,00	9,95
Total Demanda [MW]	18,91	30,00	70,00

Tabla A.4.19: Potencias resultantes del Método de Kirschen, Allan y Strbac sobre la red de prueba. Elaboración propia.

Asignación de pérdidas a productores			
Generación [MW]		Contribución a la Demanda Total [MW]	Pérdidas por Generador [MW]
Gen 1	60,00	59,41	0,59
Gen 3	50,00	49,55	0,45
Gen 6	10,00	9,95	0,05
Total [MW]	120,00	118,91	1,09

Tabla A.4.20: Responsabilidad del parque generador en el uso del sistema de transmisión. Elaboración propia.

En términos generales, los resultados anteriores exponen el nivel de participación que cada agente productor tiene sobre el SEP de la *Figura A.4.4*; evidenciando las contribuciones sobre cada *área común de generación*, así como el costo físico que los generadores deben asumir para transportar la energía hacia dichas zonas (ya que, como se mencionó en el Capítulo 4, este método sólo asigna las pérdidas sobre las fuentes del sistema).

La *Tabla A.4.18*, por ejemplo, muestra que cerca del 42% de las cargas y el *flujo externo* del *área común 3* se abastece de las inyecciones de G_1, mientras que el *área común 6* consume $28{,}10[MW]$

procedentes del mismo generador, de los cuales $27{,}94[MW]$ están destinados a abastecer a la demanda (ver *Tabla A.4.19*).

También se puede ver que G_3, por su lado, aporta con el $58{,}15\%$ de la potencia demandada por las cargas y líneas del *área común 3*, porción que posteriormente es aprovechada para constituir el *flujo externo* que se importará desde dicha región hacia el *área común 6*, cuyos flujos son alimentados en un $85{,}79\%$ con el suministro de los generadores G_1 y G_3.

La *Tabla A.4.20*, en tanto, exhibe el grado de uso que cada generador ejerce sobre el sistema de transporte, siendo los generadores G_1 y G_3 los actores que producen mayores pérdidas en la red; en parte porque son quienes mayor producción energética tienen y, a su vez, porque no tienen consumos inmediatos en las barras a las que están conectados. Al respecto, el generador G_6 produce sólo $10[MW]$, pero son consumidos casi íntegramente por la carga $\mathcal{D}_6$ debido a la conexión directa que existe entre ellos.

Para terminar, y en conformidad a lo que declaran los autores [2], es bueno acotar que para conocer el grado de utilización que tiene un generador específico sobre las líneas de transmisión que unen *áreas comunes* distintas, se debe realizar un análisis basado en las contribuciones del parque generador sobre las *áreas comunes* de menor *rango*; pero eso escapa de las finalidades de este trabajo.

8.4.9 Código MATLAB Ej. SEP de prueba – M. de Kirschen, Allan y Strbac

```
%Ejemplo Kirschen, Allan & Strbac 6 barras

%--------------------------------Lectura de los datos del SEP-------------------------------
%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
hojactiva=actarch.Activesheet;
Range=get(hojactiva,'Range','C3','E8');
%Matriz de array para GD
P1=Range.value;
invoke(oparch,'Close');
invoke(actarch,'Quit');
%Matriz numérica para Gen y Dem
[M N]=size(P1);
inarch=cat(2,P1(:));
GD=reshape(inarch,M,N);
Barras=GD(:,1);
Gen=GD(:,2);
Dem=GD(:,3);

%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
hojactiva=actarch.Activesheet;
Range=get(hojactiva,'Range','G4','J12');
```

```
%Matriz de array para Red
P2=Range.value;
invoke(oparch,'Close');
invoke(actarch,'Quit');
%Matriz numérica para Red
[H I]=size(P2);
inarch=cat(2,P2(:));
Red=reshape(inarch,H,I);


Lineas=[];%Muestra la numeración de los extremos de cada línea (barras o nodos del SEP)
Flujos=[];%Flujos de potencia asociados a cada línea del SEP

for i=1:size(Red,1)
    for j=1:size(Red,2)
        if j<3
        Lineas(i,j)=Red{i,j};
        elseif j>=3
        Flujos(i,j-2)=Red{i,j};
        end
    end
end

%-----------------------------Construcción de variables de entrada------------------------

D=[];

for i=1:size(Barras,1)
    for j=1:size(Barras,1)
        if Gen{i,1}>0
         if i==1
         D(i,j)=j;
         elseif i==3&&j>2
         D(i,j)=j;
         elseif i==6&&j>5
         D(i,j)=j;
         else
         D(i,j)=0;
         end
        end
    end
end

for i=1:size(Gen,1)
Dom{i}=D(i,:);%Muestra el "dominio de generación" de cada Generador del SEP conectado a la
"i" barra
end

ACom{1}=cat(2,Dom{1}(1,1),Dom{1}(1,2));
ACom{3}=cat(2,Dom{3}(1,3),Dom{3}(1,4),Dom{3}(1,5));
ACom{6}=cat(2,Dom{6}(1,6));


for i=1:size(Gen,1)
    if find(ACom{i}==1)
    Rango{i}=1; %Rango "área común de generación" Gen 1
    elseif find(ACom{i}==3)
    Rango{i}=2; %Rango "área común de generación" Gen 3
    elseif find(ACom{i}==6)
    Rango{i}=3; %Rango "área común de generación" Gen 6
    end
end


CargaACom{1}=Dem{1,1}+Dem{2,1};%Consumo total en "área común de generación" Gen 1
CargaACom{3}=Dem{4,1}+Dem{5,1};%Consumo total en "área común de generación" Gen 2
CargaACom{6}=Dem{6,1};%Consumo total en "área común de generación" Gen 1


%-----------------------------Fin construcción de variables de entrada----------------------
```

```
%--------------------------------Método recursivo del sistema-----------------------------

Enlace1_3=Flujos(2,1)+Flujos(3,1);%Flujo de potencia total entre "áreas comunes" 1 y 3
Enlace3_6=Flujos(8,1)+Flujos(9,1);%Flujo de potencia total entre "áreas comunes" 3 y 6
Enlace1_6=Flujos(4,1);%Flujo de potencia total entre "áreas comunes" 1 y 6

I_1=Gen{1,1};%Flujo interno "área común de generación 1"
I_3=Gen{3,1}+Enlace1_3;%Flujo interno "área común de generación 3"
I_6=Gen{6,1}+Enlace3_6+Enlace1_6;%Flujo interno "área común de generación 6"

fG1_11=Gen{1,1};%Flujo aportado en el "área común 1" por Gen 1: Nodo raiz
fG3_33=Gen{3,1};%Flujo aportado en el "área común 3" por Gen 3
fG6_66=Gen{6,1};%Flujo aportado en el "área común 6" por Gen 6

cG1_1=(fG1_11/I_1);%Contribución relativa de Gen 1 sobre las cargas y el flujo externo del
"área común 1"
cG3_3=(fG3_33/I_3);%Contribución relativa de Gen 3 sobre las cargas y el flujo externo del
"área común 3"
cG6_6=(fG6_66/I_6);%Contribución relativa de Gen 6 sobre las cargas y el flujo externo del
"área común 6"
fG1_13=cG1_1*Enlace1_3;%Contribución de Gen1 al flujo que transita desde "área común 1" a
"área común 3"
cG1_3=(fG1_13/I_3);%Contribución relativa de Gen 1 sobre las cargas y el flujo externo del
"área común 3"
fG1_16=cG1_1*Enlace1_6+cG1_3*Enlace3_6;%Contribución de Gen1 al flujo que transita desde
"área común 1" a "área común 6"
fG3_36=cG3_3*Enlace3_6;%Contribución de Gen3 al flujo que transita desde "área común 3" a
"área común 6"
cG1_6=(fG1_16/I_6);%Contribución relativa de Gen 1 sobre las cargas y el flujo externo del
"área común 6"
cG3_6=(fG3_36/I_6);%Contribución relativa de Gen 3 sobre las cargas y el flujo externo del
"área común 6"

fG1_CargaACom{1}=cG1_1*CargaACom{1};%Contribución de Gen1 a la demanda del "área común 1"

fG1_CargaACom{3}=cG1_3*CargaACom{3};%Contribución de Gen1 a la demanda del "área común 3"
fG3_CargaACom{3}=cG3_3*CargaACom{3};%Contribución de Gen3 a la demanda del "área común 3"

fG1_CargaACom{6}=cG1_6*CargaACom{6};%Contribución de Gen1 a la demanda del "área común 6"
fG3_CargaACom{6}=cG3_6*CargaACom{6};%Contribución de Gen3 a la demanda del "área común 6"
fG6_CargaACom{6}=cG6_6*CargaACom{6};%Contribución de Gen3 a la demanda del "área común 6"

%--------------------------------Fin Método recursivo del sistema-------------------------


%----------------------Registro de los resultados del Método de KA&S en 6 barras------------

%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
set(actarch,'Visible',1);%Activa la hoja de Excel
hojactiva=actarch.Activesheet;
hojaRange=get(hojactiva,'Range','D22');
hojaRange1=get(hojactiva,'Range','E22');
hojaRange2=get(hojactiva,'Range','E23');
hojaRange3=get(hojactiva,'Range','F22');
hojaRange4=get(hojactiva,'Range','F23');
hojaRange5=get(hojactiva,'Range','F24');
hojaRange6=get(hojactiva,'Range','D30');
hojaRange7=get(hojactiva,'Range','E30');
hojaRange8=get(hojactiva,'Range','E31');
hojaRange9=get(hojactiva,'Range','F30');
hojaRange10=get(hojactiva,'Range','F31');
hojaRange11=get(hojactiva,'Range','F32');
set(hojaRange,'Value',cG1_1);
set(hojaRange1,'Value',cG1_3);
set(hojaRange2,'Value',cG3_3);
set(hojaRange3,'Value',cG1_6);
set(hojaRange4,'Value',cG3_6);
```

```
set(hojaRange5,'Value',cG6_6);
set(hojaRange6,'Value',fG1_CargaACom{1});
set(hojaRange7,'Value',fG1_CargaACom{3});
set(hojaRange8,'Value',fG3_CargaACom{3});
set(hojaRange9,'Value',fG1_CargaACom{6});
set(hojaRange10,'Value',fG3_CargaACom{6});
set(hojaRange11,'Value',fG6_CargaACom{6});
invoke(oparch,'Close');
invoke(actarch,'Quit');
```

8.4.10 Ejemplo del Método de Prorrateo o Estampillado Postal

Con el fin de mostrar cómo usar el Método de Prorrateo o Estampillado Postal en un SEP sencillo, se tomará nuevamente el sistema propuesto en [2] (con flujo de carga en estado permanente obtenido en DigSilent PF mediante elaboración propia) y cuyas principales características se informan a continuación:

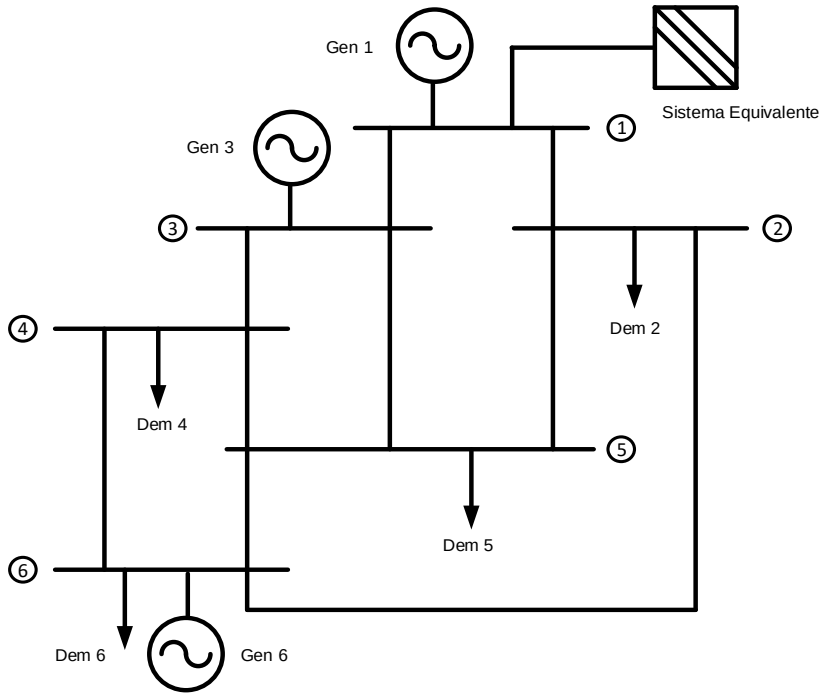


Figura A.4.7: SEP de prueba utilizado en [2] para ejemplificar el Método de Kirschen *et al*. Elaboración propia.

Barra	Generación [MW]	Demanda [MW]
1	60,00	-1,09
2	0,00	20,00
3	50,00	0,00
4	0,00	10,00
5	0,00	20,00
6	10,00	70,00
Total [MW]	120,00	118,91
Pérdidas [MW]	1,09	

Tabla 4.21: Datos de inyección y retiro de los agentes del SEP de la *Figura A.4.7*.

Línea de Transmisión		Flujo de Potencia [MW]	
Barra "i"	Barra "j"	Pij_ret(i)	Pij_iny(j)
1	2	48,35	48,11
1	3	12,74	12,68
2	5	23,25	23,12
2	6	4,86	4,81
3	4	49,47	49,37
3	5	13,21	13,13
4	5	18,14	18,06
4	6	21,22	21,04
5	6	34,30	34,15
Total [MW]		225,55	224,46
Pérdidas [MW]		1,09	

Tabla 4.22: Flujo de carga AC arrojado por el software DigSilent PF (archivo "pfd" disponible). Elaboración propia.

De los datos agrupados se observa que las pérdidas totales atribuidas al transporte de la energía son próximas a $1{,}09[MW]$. Por otra parte, las barras en las que se efectúa el retiro de potencia son fundamentalmente cuatro (barra 2, barra 4, barra 5 y barra 6) y su consumo es conocido; tal como se presenta en la *Tabla 4.21*.

Con lo anterior, es posible calcular las pérdidas eléctricas que serán asignadas a cada carga en virtud de la magnitud de sus retiros, atribuyéndole un cargo físico por la potencia desplazada desde las fuentes de producción hacia los centros de consumo.

Se consigue:

$$\mathcal{D}_{total} \cong 118{,}91[MW] \ ; \ P_{loss_{total}} \cong \ 1{,}09[MW]$$

$$\Rightarrow P_{loss_2} = \frac{\mathcal{D}_2}{\mathcal{D}_{total}} \cdot P_{loss_{total}} = \frac{20}{118{,}91} \cdot 1{,}09 \ \Leftrightarrow \ P_{loss_2} \cong 0{,}18[MW]$$

$$\Rightarrow P_{loss_4} = \frac{\mathcal{D}_4}{\mathcal{D}_{total}} \cdot P_{loss_{total}} = \frac{10}{118{,}91} \cdot 1{,}09 \ \Leftrightarrow \ P_{loss_4} \cong 0{,}09[MW]$$

$$\Rightarrow P_{loss_5} = \frac{\mathcal{D}_5}{\mathcal{D}_{total}} \cdot P_{loss_{total}} = \frac{20}{118{,}91} \cdot 1{,}09 \ \Leftrightarrow \ P_{loss_5} \cong 0{,}18[MW]$$

$$\Rightarrow P_{loss_6} = \frac{\mathcal{D}_6}{\mathcal{D}_{total}} \cdot P_{loss_{total}} = \frac{70}{118{,}91} \cdot 1{,}09 \ \Leftrightarrow \ P_{loss_6} \cong 0{,}64[MW]$$

Los resultados anteriores enseñan que, si lo que se desea es atribuir una porción de las pérdidas totales a cada carga según la magnitud de los retiros efectuados sobre la red, la demanda $\mathcal{D}_6$ es la que más responsabilidad tiene, mientras que $\mathcal{D}_4$ es quien produce menos pérdidas en transmisión. Note que lo anterior, aún cuando no depende de la topología de la red, no deja de ser razonable si se toma en cuenta que $\mathcal{D}_6$ está conectada en la misma barra que un generador (G_6 de quien se debiese esperar que destine la mayoría de sus inyecciones a alimentar a la carga más cercana), pero debido a la magnitud de su retiro, tiene que recurrir a los generadores G_1 y G_3 para abastecerse completamente, utilizando gran parte de las líneas de transmisión, y con ello, provocando la mayoría de las pérdidas del SEP.

Bajo la misma idea, la carga $\mathcal{D}_4$ solicita menos flujos desde la red y se debiese esperar que los generadores que la alimentan sean los conectados a las barras 3 ó 6; pero esto es sólo un supuesto.

También es importante acentuar que en las barras 2 y 5 se ha asignado la misma cuantía de pérdidas, inclusive cuando la resistencia, distancia y cargabilidad de las líneas de abastecimiento pudiesen diferir.

Hay que ser enfático en decir que los párrafos precedentes son sólo interpretaciones, puesto que el Método de Prorrateo o Estampillado Postal da cuenta de la proporción de uso que hace cada demanda del SEP en relación al uso total de las instalaciones de transmisión, de manera que es imposible saber cuáles son las líneas utilizadas por cada agente o quiénes son los proveedores que abastecen a una carga puntual. Con lo anterior, si bien esta metodología permite resolver el problema de pérdidas a través de operaciones sencillas, por sí sola es insuficiente para resolver el problema de trazabilidad, debiendo complementarse, necesariamente, con otra técnica que pueda completar la *cadena de suministro*.

8.5 Anexos Capítulo 5

8.5.1 Datos de entrada para la aplicación de Flujos Gruesos de Bialek en modelo simplificado del SEN sujeto al estado de la red del 28 de febrero de 2018 a las 14:00 hrs.

Barra	Generación [MW]	Demanda [MW]
1	253,07	0,00
2	62,56	0,00
3	212,08	0,00
4	83,07	168,39
5	222,00	0,00
6	262,05	0,00
7	48,27	0,00
8	818,00	0,00
9	1445,00	812,60
10	149,57	179,25
11	0,00	1979,16
12	117,00	0,00
13	190,33	1368,05
14	210,80	0,00
15	215,00	0,00
16	556,90	17,13
Total [MW]	4845,70	4524,58
Pérdidas [MW]	321,12	

Tabla A.5.1: Datos empíricos de entrada para el modelo reducido del SEN mostrado en la *Figura 5.1*. Elaboración propia.

Línea de Transmisión		Flujo de Potencia [MW]	
Barra "i"	Barra "j"	Pij_ret(i)	Pij_iny(j)
1	2	253,07	249,48
2	3	7,57	7,57
2	4	304,47	299,17
3	4	219,65	214,93
4	5	428,78	424,40
5	6	646,40	637,81
6	7	899,86	844,42
7	8	892,69	846,52
8	9	1164,31	1162,17

8	10	500,21	497,86
9	10	1794,57	1790,03
10	11	1284,31	1170,14
10	13	973,90	973,83
12	11	117,00	116,98
13	11	758,87	692,04
14	13	145,34	144,12
16	13	819,66	818,65
14	15	65,46	65,29
15	16	280,29	279,89
Total [MW]		11556,39	11235,28
Pérdidas [MW]		321,12	

Tabla A.5.2: Resultado del flujo de carga del SEN reducido de la Figura 5.1 (hecho en DigSilent PF). Elaboración propia.

	Balance Nodal Neto [MW]	
Barra	Inyección	Retiro
1	253,07	253,07
2	312,04	312,04
3	219,65	219,65
4	597,17	597,17
5	646,40	646,40
6	899,86	899,86
7	892,69	892,69
8	1664,52	1664,52
9	2607,17	2607,17
10	2437,46	2437,46
11	1979,16	1979,16
12	117,00	117,00
13	2126,92	2126,92
14	210,80	210,80
15	280,29	280,29
16	836,79	836,79

Tabla A.5.3: Sumatoria de inyecciones y retiros por barra. Elaboración propia.

8.5.2 Resultados Flujos Gruesos de Bialek en modelo simplificado del SEN sujeto al estado de la red del 28 de febrero de 2018 a las 14:00 hrs.

Barra	Potencia Nodal Gruesa [MW]	Demanda Gruesa [MW]	Contribuciones de cada Generador Grueso sobre Demanda Gruesa [MW]			
1	253,07	0,00	0,00	0,00	0,00	0,00
2	315,63	0,00	0,00	0,00	0,00	0,00
3	219,74	0,00	0,00	0,00	0,00	0,00
4	610,78	172,23	71,36	17,64	59,80	23,42
5	660,55	0,00	0,00	0,00	0,00	0,00
6	922,60	0,00	0,00	0,00	0,00	0,00
7	970,87	0,00	0,00	0,00	0,00	0,00
8	1788,87	0,00	0,00	0,00	0,00	0,00
9	2696,30	840,38	39,62	9,79	33,20	13,00
10	2543,06	187,02	10,45	2,58	8,76	3,43
11	2231,90	2231,90	95,13	23,52	79,72	31,23
12	117,00	0,00	0,00	0,00	0,00	0,00
13	2171,99	1397,04	36,52	9,03	30,60	11,99
14	210,80	0,00	0,00	0,00	0,00	0,00
15	280,46	0,00	0,00	0,00	0,00	0,00
16	837,36	17,14	0,00	0,00	0,00	0,00
Total [MW]	-	4845,698	253,07	62,56	212,08	83,07
			G1 [MW]	G2 [MW]	G3 [MW]	G4 [MW]

Tabla A.5.4 – parte 1

Barra	Potencia Nodal Gruesa [MW]	Demanda Gruesa [MW]	Contribuciones de cada Generador Grueso sobre Demanda Gruesa [MW]			
1	253,07	0,00	0,00	0,00	0,00	0,00
2	315,63	0,00	0,00	0,00	0,00	0,00
3	219,74	0,00	0,00	0,00	0,00	0,00
4	610,78	172,23	0,00	0,00	0,00	0,00
5	660,55	0,00	0,00	0,00	0,00	0,00
6	922,60	0,00	0,00	0,00	0,00	0,00
7	970,87	0,00	0,00	0,00	0,00	0,00
8	1788,87	0,00	0,00	0,00	0,00	0,00
9	2696,30	840,38	48,40	57,13	10,52	178,34
10	2543,06	187,02	12,77	15,07	2,78	47,04
11	2231,90	2231,90	116,22	137,19	25,27	428,23
12	117,00	0,00	0,00	0,00	0,00	0,00
13	2171,99	1397,04	44,61	52,66	9,70	164,39
14	210,80	0,00	0,00	0,00	0,00	0,00
15	280,46	0,00	0,00	0,00	0,00	0,00
16	837,36	17,14	0,00	0,00	0,00	0,00
Total [MW]	-	4845,698	222,00	262,05	48,27	818,00
			G5 [MW]	G6 [MW]	G7 [MW]	G8 [MW]

Tabla A.5.4 – parte 2

Barra	Potencia Nodal Gruesa [MW]	Demanda Gruesa [MW]	Contribuciones de cada Generador Grueso sobre Demanda Gruesa [MW]			
1	253,07	0,00	0,00	0,00	0,00	0,00
2	315,63	0,00	0,00	0,00	0,00	0,00
3	219,74	0,00	0,00	0,00	0,00	0,00
4	610,78	172,23	0,00	0,00	0,00	0,00
5	660,55	0,00	0,00	0,00	0,00	0,00
6	922,60	0,00	0,00	0,00	0,00	0,00
7	970,87	0,00	0,00	0,00	0,00	0,00
8	1788,87	0,00	0,00	0,00	0,00	0,00
9	2696,30	840,38	450,38	0,00	0,00	0,00
10	2543,06	187,02	73,14	11,00	0,00	0,00
11	2231,90	2231,90	665,86	100,13	0,00	117,00
12	117,00	0,00	0,00	0,00	0,00	0,00
13	2171,99	1397,04	255,62	38,44	0,00	0,00
14	210,80	0,00	0,00	0,00	0,00	0,00
15	280,46	0,00	0,00	0,00	0,00	0,00
16	837,36	17,14	0,00	0,00	0,00	0,00
Total [MW]	-	4845,698	1445,00	149,57	-	117,00
			G9 [MW]	G10 [MW]	G11 [MW]	G12 [MW]

Tabla A.5.4 – parte 3

Barra	Potencia Nodal Gruesa [MW]	Demanda Gruesa [MW]	Contribuciones de cada Generador Grueso sobre Demanda Gruesa [MW]			
1	253,07	0,00	0,00	0,00	0,00	0,00
2	315,63	0,00	0,00	0,00	0,00	0,00
3	219,74	0,00	0,00	0,00	0,00	0,00
4	610,78	172,23	0,00	0,00	0,00	0,00
5	660,55	0,00	0,00	0,00	0,00	0,00
6	922,60	0,00	0,00	0,00	0,00	0,00
7	970,87	0,00	0,00	0,00	0,00	0,00
8	1788,87	0,00	0,00	0,00	0,00	0,00
9	2696,30	840,38	0,00	0,00	0,00	0,00
10	2543,06	187,02	0,00	0,00	0,00	0,00
11	2231,90	2231,90	67,91	74,73	75,14	194,63
12	117,00	0,00	0,00	0,00	0,00	0,00
13	2171,99	1397,04	122,42	134,73	135,46	350,87
14	210,80	0,00	0,00	0,00	0,00	0,00
15	280,46	0,00	0,00	0,00	0,00	0,00
16	837,36	17,14	0,00	1,34	4,40	11,40
Total [MW]	-	4845,698	190,33	210,80	215,00	556,90
			G13 [MW]	G14 [MW]	G15 [MW]	G16 [MW]

Tabla A.5.4 – parte 4

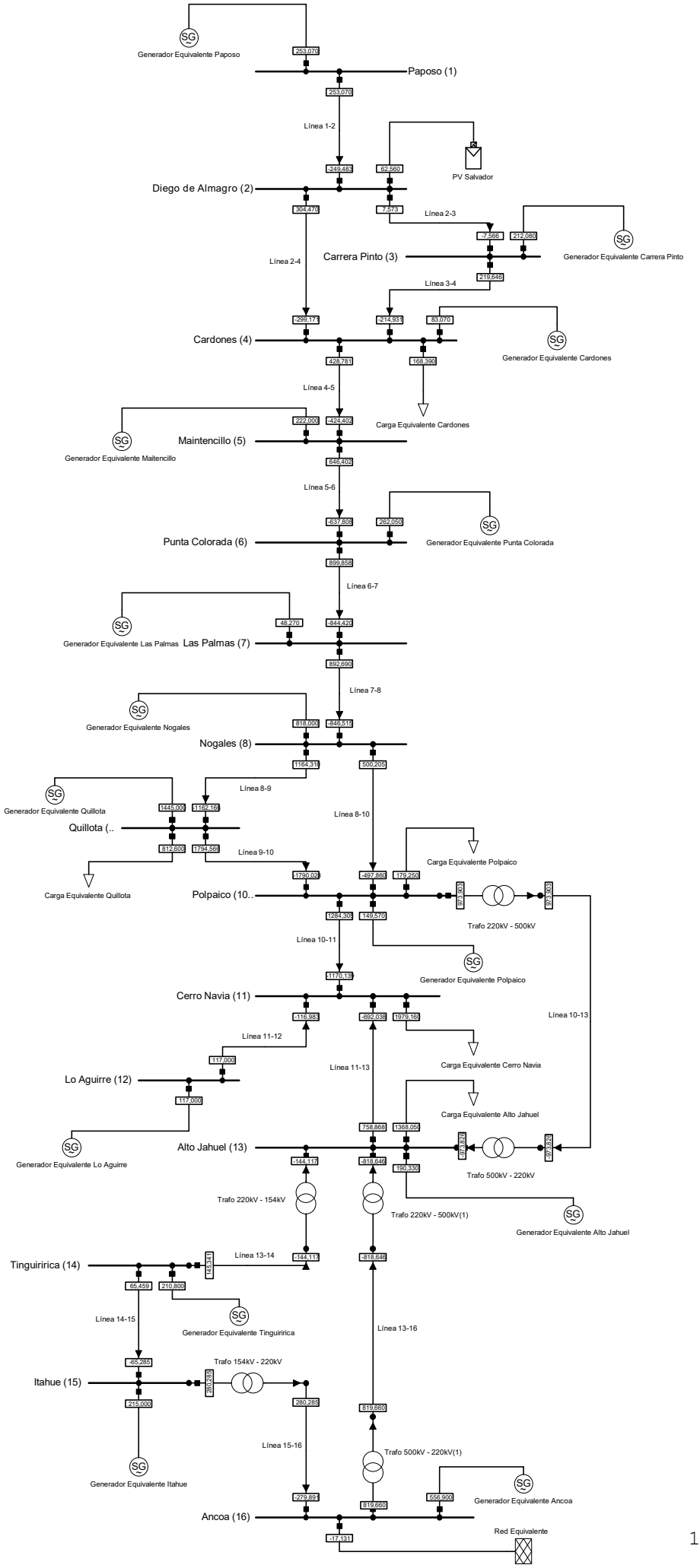
8.5.3 Código MATLAB en SEN reducido - Método de Bialek Flujos Gruesos

```
%Ejemplo Bialek SEN reducido - SEP 16 barras "Flujos Gruesos"

%--------------------------------Lectura de los datos del SEP-----------------------------

%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
hojactiva=actarch.Activesheet;
Range=get(hojactiva,'Range','D5','E20');
%Matriz de array para GD
P1=Range.value;
invoke(oparch,'Close');
invoke(actarch,'Quit');
%Matriz numérica para Gen y Dem
[M N]=size(P1);
inarch=cat(2,P1(:));
GD=reshape(inarch,M,N);
Gen=GD(:,1);
Dem=GD(:,2);

%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
hojactiva=actarch.Activesheet;
Range=get(hojactiva,'Range','G6','J24');
%Matriz de array para Red
P2=Range.value;
invoke(oparch,'Close');
invoke(actarch,'Quit');
%Matriz numérica para Red
[H I]=size(P2);
inarch=cat(2,P2(:));
Red=reshape(inarch,H,I);

Lineas=[];
Flujos=[];

for i=1:size(Red,1)
    for j=1:size(Red,2)
        if j<3
        Lineas(i,j)=Red{i,j};
        elseif j>=3
        Flujos(i,j-2)=Red{i,j};
        end
    end
end


Nodoret=Lineas(:,1);
Nodoiny=Lineas(:,2);

%--------------------------------Construcción matrices de entrada---------------------------

P_G=[]; %muestra la matriz de generación del SEP "[P_G]"

for i=1:size(Gen,1)
    for j=1:size(Gen,1)
        if i==j
        P_G(i,j)=Gen{i,1};
        else
        P_G(i,j)=0;
        end
    end
end
```

```
P_D=[];%muestra la matriz de demandas de cada barra del SEP "[P_D]"

for i=1:size(Dem,1)
    for j=1:size(Dem,1)
        if i==j
        P_D(i,j)=Dem{i,1};
        else
        P_D(i,j)=0;
        end
    end
end


P_ret_barra=[]; % matriz de flujos retirados desde la barra "fila" por la barra "columna".
Flujos por las líneas. Potencias grandes.

for i=1:size(Gen,1)
    for j=1:size(Gen,1)

        for k=1:size(Lineas,1)
            for l=1:size(Lineas,2)

                if l==1
                    if i==Lineas(k,l)&&j==Lineas(k,l+1)

                        if Flujos(k,l)>Flujos(k,l+1)
                        P_ret_barra(i,j)=Flujos(k,l);
                       elseif Flujos(k,l+1)>Flujos(k,l)
                        P_ret_barra(i,j)=Flujos(k,l+1);
                        end
                elseif l==size(Lineas,2)
                    if i==Lineas(k,l-1)&&j==Lineas(k,l)

                        if Flujos(k,l-1)>Flujos(k,l)
                        P_ret_barra(i,j)=Flujos(k,l-1);
                       elseif Flujos(k,l)>Flujos(k,l-1)
                        P_ret_barra(i,j)=Flujos(k,l);
                        end
                    end
                    end
                end
            end
        end
    end
end


P_iny_barra=[]; %matriz de flujos inyectados en la barra "columna" proveninetes desde la
barra "fila". Flujos por las líneas. Potencias chicas.

for i=1:size(Gen,1)
    for j=1:size(Gen,1)

        for k=1:size(Lineas,1)
            for l=1:size(Lineas,2)

                if l==1
                    if i==Lineas(k,l)&&j==Lineas(k,l+1)

                        if Flujos(k,l)<Flujos(k,l+1)
                        P_iny_barra(i,j)=Flujos(k,l);
                       elseif Flujos(k,l+1)<Flujos(k,l)
                        P_iny_barra(i,j)=Flujos(k,l+1);
                        end

                elseif l==size(Lineas,2)
                    if i==Lineas(k,l-1)&&j==Lineas(k,l)

                        if Flujos(k,l-1)<Flujos(k,l)
```

```
                        P_iny_barra(i,j)=Flujos(k,l-1);
                       elseif Flujos(k,l)<Flujos(k,l-1)
                        P_iny_barra(i,j)=Flujos(k,l);
                        end

                    end
                    end
                end
            end
        end
    end
end


Pnodo_G=zeros(size(Gen,1),size(Gen,1)); %matriz que agrupa las potencias totales a través de
los nodos de generación

for i=1:size(Gen,1) %Filas matriz [Pnodo_G]
    for j=1:size(Gen,1) %Columnas matriz [Pnodo_G]
        if i==j&&i<11
            Pnodo_G(i,j)=P_G(i,j)+sum(P_iny_barra(:,j));
        elseif i==j&&i>11
            Pnodo_G(i,j)=P_G(i,j)+sum(P_iny_barra(:,j));
        end
    end
end


Pnodo_D=zeros(size(Gen,1),size(Gen,1)); %matriz que agrupa las potencias totales a través de
los nodos de demanda

for i=1:size(Gen,1) %Filas matriz [Pnodo_D]
    for j=1:size(Gen,1) %Columnas matriz [Pnodo_D]
        if i==j&&i==11
            Pnodo_D(i,j)=P_D(i,j)+sum(P_ret_barra(i,:));
        end
    end
end


P_nodal_total=Pnodo_G+Pnodo_D; %muestra la potencia nodal neta a través de cada barra del
SEP bajo Flujos Gruesos


% -------------------------------Fin matrices de entrada----------------------------------

%-----------------------------Construcción de la matriz "[Au]"----------------------------


Au=[]; %matriz de coeficientes "[Au]"

for i=1:size(Gen,1) %Filas matriz [Au]
    for j=1:size(Gen,1) %Columnas matriz [Au]
        if  j==i
            Au(i,j)=1;
        else
            Au(i,j)=-abs(P_ret_barra(j,i))/P_nodal_total(j,j);
        end
    end
end


% --------------------------Fin construcción de la matriz "[Au]"--------------------------

Au_inv=Au^(-1); %inversa de la matriz de coeficientes, denotada por "[Au]^(-1)"

PG=zeros(size(P_G,1),1); % vector de generación "[PG]"

for i=1:size(P_G,1)
    PG(i,1)=P_G(i,i);
end
```

```
% -------------------------------Despeje de demandas gruesas-------------------------------

P_nodal_gruesa=(Au^(-1))*PG;

% % ---------------------------Fin de despeje de demandas gruesas---------------------------


% % -------------------------Resultados Flujos Gruesos J.Bialek----------------------------

Pcarga_gruesa=[]; %potencia demandada en las cargas bajo el uso de factores gruesos
"P_nodal_gruesa(i,1)"

for i=1:size(Dem,1)
    Pcarga_gruesa(i,1)=(Dem{i,1}/P_nodal_total(i,i))*(P_nodal_gruesa(i,1));
end


Aportes_G=[]; %Contribuciones de energía del Generador "columna" en la demanda gruesa de la
Carga "fila"

for i=1:size(Au_inv,1)
    for j=1:size(Au_inv,2)
        Aportes_G(i,j)=(Dem{i,1}/P_nodal_total(i,i))*Au_inv(i,j)*PG(j,1);
    end
end


%-------------------------Registro de los resultados del Método de Bialek-------------------

%Apertura de la hoja de Excel
[file,path] = uigetfile({'*.xlsx','Excel(*.xlsx)'},'Matrices de entrada');
target=strcat(path,file);
actarch=actxserver('Excel.Application');
oparch=invoke(actarch.workbooks,'Open',target);
set(actarch,'Visible',1);%Activa la hoja de Excel
hojactiva=actarch.Activesheet;
hojaRange=get(hojactiva,'Range','D32','D47');
hojaRange1=get(hojactiva,'Range','E32','E47');
hojaRange2=get(hojactiva,'Range','F32','U47');
set(hojaRange,'Value',P_nodal_gruesa);%Escribir Factores Gruesos sobre hoja Excel
set(hojaRange1,'Value',Pcarga_gruesa);%Escribir Demandas Gruesas sobre hoja Excel
set(hojaRange2,'Value',Aportes_G);%Escribir las contribuciones de flujo de los Generadores
sobre hoja Excel
invoke(oparch,'Close');
invoke(actarch,'Quit');
```



Generador Equivalente Paposo
253,070
Paposo (1)
253,070
Línea 1-2
PV Salvador
-249,483
62,560
Diego de Almagro (2)
304,470
7,573
Línea 2-3
-7,566
212,080
Generador Equivalente Carrera Pinto
Carrera Pinto (3)
Línea 2-4
219,646
Línea 3-4
-299,171
-214,931
83,070
Cardones (4)
Generador Equivalente Cardones
428,781
168,390
Línea 4-5
Carga Equivalente Cardones
222,000
-424,402
Maintencillo (5)
Generador Equivalente Maitencillo
646,402
Línea 5-6
-637,808
262,050
Punta Colorada (6)
Generador Equivalente Punta Colorada
899,858
Línea 6-7
48,270
-844,420
Generador Equivalente Las Palmas
Las Palmas (7)
892,690
Línea 7-8
Generador Equivalente Nogales
818,000
-846,515
Nogales (8)
1164,310
500,205
Línea 8-9
Generador Equivalente Quillota
1445,000
-1162,164
Línea 8-10
Quillota (..
812,600
1794,566
Línea 9-10
Carga Equivalente Polpaico
-1790,028
-497,860
179,250
Carga Equivalente Quillota
Polpaico (10..
973,903
973,903
1284,305
149,570
Trafo 220kV - 500kV
Línea 10-11
Generador Equivalente Polpaico
-1170,131
Cerro Navia (11)
-116,983
-692,038
1979,160
Línea 10-13
Línea 11-12
117,000
Carga Equivalente Cerro Navia
Línea 11-13
Lo Aguirre (12)
117,000
Carga Equivalente Alto Jahuel
758,868
1368,050
Generador Equivalente Lo Aguirre
Alto Jahuel (13)
-973,826
-973,826
-144,117
-818,646
190,330
Trafo 500kV - 220kV
Trafo 220kV - 154kV
Trafo 220kV - 500kV(1)
Generador Equivalente Alto Jahuel
Tinguiririca (14)
145,341
Línea 13-14
-144,117
-818,646
65,459
210,800
Línea 14-15
Generador Equivalente Tinguiririca
Línea 13-16
-65,285
Trafo 154kV - 220kV
Itahue (15)
280,285
215,000
280,285
819,660
Línea 15-16
Trafo 500kV - 220kV(1)
Generador Equivalente Itahue
-279,891
819,660
556,900
Generador Equivalente Ancoa
Ancoa (16)
-17,131
Red Equivalente