

Proyecto Fin de Máster Máster de Ingeniería Electrónica, Robótica y Auto- mática

Diseño de una Plataforma de Gestión de comunidades energéticas con Blockchain y Control Predictivo Distribuido

Autor: Manuel Sivianes Castaño

Tutor: Carlos Bordons Alba

**Dpto. Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2023



Proyecto Fin de Máster
Máster de Ingeniería Electrónica, Robótica y Automática

Diseño de una Plataforma de Gestión de comunidades energéticas con Blockchain y Control Predictivo Distribuido

Autor:

Manuel Sivianes Castaño

Tutor:

Carlos Bordons Alba

Catedrático

Dpto. Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2023

Proyecto Fin de Máster: Diseño de una Plataforma de Gestión de comunidades energéticas con Blockchain y Control Predictivo Distribuido

Autor: Manuel Sivianes Castaño
Tutor: Carlos Bordons Alba

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Resumen

Este Trabajo de Fin de Máster (TFM) se adentra en la comprensión y aplicación de la tecnología blockchain en microrredes o comunidades energéticas, con un enfoque particular en los conceptos fundamentales de esta tecnología y, específicamente, en la red de Ethereum, reconocida por su madurez en términos de contratos inteligentes. El objetivo principal es presentar y desarrollar los resultados expuestos en [67] en el contexto de comunidades energéticas gestionadas por un algoritmo Tree-based MPC (TBMPC) distribuido coordinado por un contrato inteligente.

En contraste con otros enfoques que emplean TBMPC para abordar incertidumbres en contextos similares, este trabajo destaca por su enfoque distribuido, reemplazando la capa superior convencional, que suele implicar una autoridad central, por un contrato inteligente en Ethereum. Este cambio elimina la necesidad de terceros en la gestión de datos y aprovecha la plataforma OpenZeppelin Defender para supervisar eventos en blockchain y automatizar funciones de contratos inteligentes.

En cuanto a la implementación distribuida, se ha optado por el algoritmo Feasible Cooperation-based MPC (FC-MPC), que presenta ventajas significativas en comparación con otros algoritmos utilizados en comunidades energéticas. A diferencia de métodos como ADMM, el FC-MPC garantiza que todas las soluciones intermedias sean factibles para el sistema en su conjunto, incluso si el algoritmo finaliza antes de alcanzar la convergencia. Esta característica, junto con su robustez y aplicabilidad a una variedad de problemas, destaca al FC-MPC como una elección eficaz para la gestión distribuida en comunidades energéticas.

En resumen, este TFM profundiza en la aplicación de blockchain, especialmente en Ethereum, para gestionar comunidades energéticas mediante un enfoque distribuido con el uso innovador de contratos inteligentes y el algoritmo FC-MPC. Los resultados y enfoques presentados abren nuevas perspectivas para mejorar la eficiencia y la gestión descentralizada en el contexto de la energía.

Abstract

This Master's Thesis delves into the understanding and application of blockchain technology in microgrids or energy communities, with a specific focus on the fundamental concepts of this technology and, notably, the Ethereum network, renowned for its maturity in terms of smart contracts. The main objective is to present and develop the results outlined in [67] in the context of energy communities managed by a distributed TBMPC algorithm coordinated by a smart contract.

In contrast to other approaches employing TBMPC to address uncertainties in similar contexts, this work stands out for its distributed approach, replacing the conventional upper layer, which usually involves a central authority, with a smart contract on Ethereum. This shift eliminates the need for third parties in data management and leverages the OpenZeppelin Defender platform to monitor events on the blockchain and automate smart contract functions.

Regarding the distributed implementation, the Feasible Cooperation-based MPC (FC-MPC) algorithm has been chosen, offering significant advantages over other algorithms commonly used in energy communities. Unlike methods like ADMM, FC-MPC ensures that all intermediate solutions are feasible for the entire system, even if the algorithm terminates before reaching convergence. This feature, coupled with its robustness and applicability to a broader range of problems, positions FC-MPC as an effective choice for distributed management in energy communities.

In summary, this thesis delves into the application of blockchain, particularly Ethereum, to manage energy communities through a distributed approach, employing innovative smart contract usage and the FC-MPC algorithm. The presented results and approaches open new perspectives for enhancing efficiency and decentralized management in the energy context.

Índice Abreviado

<i>Resumen</i>	I
<i>Abstract</i>	III
<i>Índice Abreviado</i>	V
<i>Notación</i>	IX
1 Introducción y objetivos	1
1.1 Objetivos	3
2 Tecnología blockchain	5
2.1 Historia de blockchain	5
2.2 Definición de blockchain	6
2.3 Funcionamiento de una Blockchain	6
2.4 Principales blockchain	8
3 Ethereum	11
3.1 Historia de Ethereum	11
3.2 Organización Autonomía Descentralizada (DAO)	13
3.3 Componentes de Ethereum	14
3.4 Principales usos de Ethereum	15
3.5 El trilema de la escalabilidad descentralizada	15
4 Gestión energética y blockchain	19
4.1 Comunidades energéticas	19
4.2 Métodos	19
4.3 Papel de blockchain	23
5 Plataforma de gestión energética diseñada	25
5.1 Formulación de la comunidad energética	25
5.2 Control predictivo particularizado en la comunidad energética	28
5.3 Control predictivo basado en árboles particularizado en la comunidad energética	29
5.4 Control predictivo distribuido particularizado en la comunidad energética	30
5.5 Desarrollo de la aplicación blockchain	32
6 Resultados y Simulaciones	37
6.1 Caso de estudio	37
6.2 Árbol de perturbaciones y parámetros de control	37
6.3 Resultados de Simulación	39
7 Conclusiones	51

<i>Índice de Figuras</i>	53
<i>Índice de Tablas</i>	55
<i>Índice de Códigos</i>	57
<i>Bibliografía</i>	59
<i>Índice alfabético</i>	63
<i>Glosario</i>	65

Índice

<i>Resumen</i>	I
<i>Abstract</i>	III
<i>Índice Abreviado</i>	V
<i>Notación</i>	IX
1 Introducción y objetivos	1
1.1 Objetivos	3
2 Tecnología blockchain	5
2.1 Historia de blockchain	5
2.2 Definición de blockchain	6
2.2.1 Componentes de una blockchain	6
2.3 Funcionamiento de una Blockchain	6
2.3.1 Interacción del usuario	6
2.3.2 Transmisión de la Transacción	7
2.3.3 Mineros	7
2.3.4 Inclusión en la Blockchain	7
2.4 Principales blockchain	8
2.4.1 Bitcoin	8
2.4.2 Ethereum	9
2.4.3 Polkadot	9
2.4.4 Chainlink	9
3 Ethereum	11
3.1 Historia de Ethereum	11
3.2 Organización Autónoma Descentralizada (DAO)	13
3.2.1 Orígenes de las DAOs	13
3.2.2 Funcionamiento de una DAO	13
3.2.3 El caso de "The DAO"	13
3.3 Componentes de Ethereum	14
3.4 Principales usos de Ethereum	15
3.5 El trilema de la escalabilidad descentralizada	15
3.5.1 Una visión un poco más detallada dentro de la escalabilidad	16
Escalabilidad on-chain: Sharding	17
Escalabilidad off-chain	17
4 Gestión energética y blockchain	19
4.1 Comunidades energéticas	19
4.2 Métodos	19
4.2.1 Formulación MPC estándar	20
4.2.2 Formulación MPC basado en escenarios	20

4.2.3	MPC basado en árboles	21
4.2.4	MPC distribuido en comunidades energéticas	21
4.3	Papel de blockchain	23
5	Plataforma de gestión energética diseñada	25
5.1	Formulación de la comunidad energética	25
	Paneles solares	25
	Baterías	26
	Vehículos eléctricos	27
	Balances de potencia	27
	Representación en espacio de estados	28
5.2	Control predictivo particularizado en la comunidad energética	28
5.3	Control predictivo basado en árboles particularizado en la comunidad energética	29
5.4	Control predictivo distribuido particularizado en la comunidad energética	30
5.5	Desarrollo de la aplicación blockchain	32
6	Resultados y Simulaciones	37
6.1	Caso de estudio	37
	6.1.1 Batería	37
	6.1.2 Paneles Solares Fotovoltaicos	37
	6.1.3 Vehículo Eléctrico	37
6.2	Árbol de perturbaciones y parámetros de control	37
6.3	Resultados de Simulación	39
	6.3.1 Simulaciones	39
7	Conclusiones	51
	<i>Índice de Figuras</i>	53
	<i>Índice de Tablas</i>	55
	<i>Índice de Códigos</i>	57
	<i>Bibliografía</i>	59
	<i>Índice alfabético</i>	63
	<i>Glosario</i>	65

Notación

\mathbb{R}	Cuerpo de los números reales
\mathbb{C}	Cuerpo de los números complejos
$\ \mathbf{v}\ $	Norma del vector \mathbf{v}
$\langle \mathbf{v}, \mathbf{w} \rangle$	Producto escalar de los vectores \mathbf{v} y \mathbf{w}
$ \mathbf{A} $	Determinante de la matriz cuadrada \mathbf{A}
$\det(\mathbf{A})$	Determinante de la matriz (cuadrada) \mathbf{A}
\mathbf{A}^\top	Transpuesto de \mathbf{A}
\mathbf{A}^{-1}	Inversa de la matriz \mathbf{A}
\mathbf{A}^\dagger	Matriz pseudoinversa de la matriz \mathbf{A}
\mathbf{A}^H	Transpuesto y conjugado de \mathbf{A}
\mathbf{A}^*	Conjugado
c.t.p.	En casi todos los puntos
c.q.d.	Como queríamos demostrar
■	Como queríamos demostrar
□	Fin de la solución
e.o.c.	En cualquier otro caso
e	número e
e^{jx}	Exponencial compleja
$e^{j2\pi x}$	Exponencial compleja con 2π
e^{-jx}	Exponencial compleja negativa
$e^{-j2\pi x}$	Exponencial compleja negativa con 2π
Re	Parte real
Im	Parte imaginaria
sen	Función seno
tg	Función tangente
arctg	Función arco tangente
$\sin^y x$	Función seno de x elevado a y
$\cos^y x$	Función coseno de x elevado a y
Sa	Función sampling
sgn	Función signo
rect	Función rectángulo
Sinc	Función sinc
$\frac{\partial y}{\partial x}$	Derivada parcial de y respecto a x
x°	Notación de grado, x grados.
$\Pr(A)$	Probabilidad del suceso A
$E[X]$	Valor esperado de la variable aleatoria X
σ_X^2	Varianza de la variable aleatoria X
$\sim f_X(x)$	Distribuido siguiendo la función densidad de probabilidad $f_X(x)$

$\mathcal{N}(m_X, \sigma_X^2)$	Distribución gaussiana para la variable aleatoria X, de media m_X y varianza σ_X^2
\mathbf{I}_n	Matriz identidad de dimensión n
$\text{diag}(\mathbf{x})$	Matriz diagonal a partir del vector \mathbf{x}
$\text{diag}(\mathbf{A})$	Vector diagonal de la matriz \mathbf{A}
SNR	Signal-to-noise ratio
MSE	Minimum square error
:	Tal que
$\stackrel{\text{def}}{=}$	Igual por definición
$\ \mathbf{x}\ $	Norma-2 del vector \mathbf{x}
$ \mathbf{A} $	Cardinal, número de elementos del conjunto \mathbf{A}
$\mathbf{x}_i, i = 1, 2, \dots, n$	Elementos i , de 1 a n , del vector \mathbf{x}
dx	Diferencial de x
\leq	Menor o igual
\geq	Mayor o igual
\backslash	Backslash
\Leftrightarrow	Si y sólo si
$x = a + 3 \underset{a=1}{=} 4$	Igual con explicación
$\frac{a}{b}$	Fracción con estilo pequeño, a/b
Δ	Incremento
$b \cdot 10^a$	Formato científico
\rightarrow_x	Tiende, con x
\mathbf{O}	Orden
TM	Trade Mark
$\mathbb{E}[x]$	Esperanza matemática de x
\mathbf{C}_x	Matriz de covarianza de \mathbf{x}
\mathbf{R}_x	Matriz de correlación de \mathbf{x}
σ_x^2	Varianza de x

1 Introducción y objetivos

El siglo XXI está siendo un momento de cambio trascendental en la historia de la energía como respuesta a las nuevas necesidades energéticas, la creciente conciencia de los desafíos del cambio climático y la urgente necesidad de reducir drásticamente las emisiones de gases de efecto invernadero [39]. En este contexto, las fuentes de energía renovable, como la solar, eólica e hidroeléctrica, se han erigido como pilares esenciales en la transición hacia un sistema energético más sostenible y limpio [44]. A medida que buscamos alternativas a los combustibles fósiles y avanzamos en la adopción de fuentes de energía más limpias, estamos cambiando no solo la manera en que generamos energía, sino también cómo optimizamos su uso.

El cambio de paradigma es evidente: en el año 2020, las fuentes de energía renovable contribuyeron significativamente al consumo de energía en la Unión Europea, superando el objetivo inicial del 20% para 2020 [23]. No obstante, aún enfrentamos el reto de cumplir con los ambiciosos objetivos de la UE para 2030 [33]. La transformación de nuestro sistema energético requiere de un enfoque adaptable, capaz de lidiar con las incertidumbres asociadas a las fuentes de energía renovables y a la creciente incorporación de recursos energéticos distribuidos. Para lograrlo, se ha impulsado la reconfiguración del sistema eléctrico, dividiéndolo en redes de pequeña escala interconectadas, donde cargas, sistemas de almacenamiento y generadores cooperan en la provisión de energía. En este contexto, las microrredes, que permiten el intercambio activo de energía entre hogares y la red eléctrica, han surgido como una solución prometedora, aunque su exposición a incertidumbres, como las condiciones meteorológicas, plantea desafíos significativos en términos de control y gestión [44, 34, 39]. Este desafío ha sido objeto de un extenso análisis debido a las importantes implicaciones económicas que conlleva. Entre las múltiples enfoques investigados, uno sobresale como especialmente eficaz en este contexto: el control predictivo basado en modelos (MPC). Esta metodología ofrece una notable aptitud para abordar la naturaleza intermitente de las fuentes de energía renovable, así como para garantizar la estabilidad de la red y gestionar las incertidumbres en la demanda energética, lo que justifica su elección en el desarrollo de este estudio [13].

El enfoque MPC engloba un amplio abanico de estrategias de control que utilizan un modelo del sistema para prever su evolución y determinar una secuencia de acciones de control, minimizando una función objetivo a lo largo de un horizonte de predicción. Sólo se implementa la primera acción de control de la secuencia, descartando las demás debido a que el problema se resuelve nuevamente en el próximo instante de tiempo, siguiendo un esquema de horizonte deslizante. Este método exhibe diversas características que lo convierten en una elección altamente atractiva para aplicaciones como el control de microrredes [13] y [62]:

1. Incorpora un mecanismo de retroalimentación que permite al controlador gestionar inexactitudes y perturbaciones.
2. Su capacidad proactiva se convierte en un pilar esencial para sistemas fuertemente expuestos a perturbaciones
3. La capacidad de lidiar con restricciones en tiempo real refuerza aún más su idoneidad en este contexto.

Dentro del ámbito de las microrredes, los enfoques de MPC que consideran perturbaciones han sido objeto de un notable interés, especialmente los enfoques estocásticos [45] y [41]. Estos métodos se dividen en dos grupos principales, como se detalla en [13]: los métodos analíticos y los basados en escenarios.

Los métodos analíticos hacen uso de conceptos estadísticos para reformular tanto las restricciones como la función objetivo. Por ejemplo, se puede encontrar un ejemplo de MPC estocástico para gestionar una microrred en [32], donde las previsiones de carga y energía eólica se modelan como distribuciones normales.

De manera similar, en el trabajo de [20], se aborda el problema mediante un esquema jerárquico de MPC estocástico para la gestión energética de una microred, incluyendo diversas fuentes de generación y carga no determinista, garantizando la satisfacción de restricciones en términos probabilísticos. Los enfoques basados en escenarios, por otro lado, parten de la premisa de que las perturbaciones desconocidas que afectan al sistema pueden describirse mediante un conjunto discreto de realizaciones, que se utilizan para calcular la secuencia óptima de entrada. Debido a su enfoque intuitivo, este método ha sido ampliamente adoptado en el contexto de las microredes, tal como se presenta en [78], donde se propone un marco de MPC basado en escenarios para la programación y control de una microred combinada de calor y energía con fuentes de energía renovable a gran escala.

La metodología de MPC de múltiples escenarios (MSMPC) produce una secuencia única de control que es factible para todos los escenarios considerados, aunque puede resultar en un rendimiento conservador en situaciones extremas, como se discute en [29]. Para evitar esto, se puede aplicar el enfoque basado en árboles para el MPC (TBMPC), que modela las posibles perturbaciones a lo largo del tiempo en una estructura de árbol, permitiendo el cálculo de una secuencia de control que refleja la estructura de cada perturbación. Este enfoque minimiza el impacto de escenarios improbables al aplicar restricciones no anticipativas, tal como se muestra en el estudio de [73], que implementa un TBMPC jerárquico y distribuido para sistemas de gestión del agua bajo perturbaciones hidroclimáticas. En un contexto similar, en el trabajo de [38], se recurre al enfoque TBMPC distribuido para sistemas de drenaje bajo perturbaciones climáticas. Finalmente, en el ámbito de las microredes, un estudio comparativo presentado en [74] analiza diversas estrategias de MPC estocástico aplicadas a una microred basada en energía renovable e hidrógeno, concluyendo que el enfoque TBMPC conlleva un coste acumulado significativamente más bajo en comparación con el enfoque MSMPC.

En microredes donde existen múltiples agentes autónomos con capacidad de toma de decisiones, y además están dispersos geográficamente, el empleo de arquitecturas de MPC distribuido se convierte en una necesidad. Estas arquitecturas otorgan a cada agente la autonomía necesaria para tomar decisiones de control, al mismo tiempo que garantizan la coordinación y optimización en todo el sistema. Esto se pone de manifiesto en el estudio de [79], donde el problema de optimización de la distribución de energía en una microred se divide en múltiples problemas MPC interconectadas que trabajan en conjunto para minimizar la función objetivo del sistema global. Además, en [17], se propone un enfoque de DMPC para microrredes, donde el problema de control óptimo se plantea como un programa lineal mixto entero, y la optimización es realizada de manera distribuida por los vecinos, dada la imposibilidad de ser resuelto centralmente. Además, en [28], se emplea un algoritmo de control basado en DMPC para optimizar el programa económico de una red de microredes interconectadas que engloba sistemas de almacenamiento de energía híbridos.

No obstante, estas arquitecturas descentralizadas y distribuidas se apoyan en el intercambio de datos e información de control entre agentes, lo que plantea preocupaciones relativas a vulnerabilidades como ataques cibernéticos. Para abordar estas cuestiones surge la tecnología blockchain, que ha ganado relevancia en los últimos años. La tecnología blockchain se originó junto con Bitcoin en 2008, con la publicación del documento técnico "Bitcoin: Un sistema de efectivo electrónico entre pares" [46]. Sus objetivos primordiales incluyen la eliminación de intermediarios en el comercio de energía y la habilitación de interacciones seguras entre pares (P2P) sin depender de una entidad central. Esto se logra mediante la aplicación de criptografía, mecanismos de consenso y una naturaleza intrínsecamente descentralizada, lo que reduce considerablemente el riesgo de ataques cibernéticos. Las blockchains se pueden definir como bases de datos distribuidas que consisten en bloques, cada uno de los cuales alberga múltiples transacciones. Estos bloques son sellados temporalmente y se enlazan al bloque anterior mediante técnicas criptográficas, dificultando en gran medida cualquier intento de interferencia por parte de atacantes. Además, los agentes tienen la libertad de unirse o abandonar la red sin afectar su rendimiento. Algunas blockchains hacen posible la ejecución de algoritmos descentralizados gracias a los contratos inteligentes, que son conjuntos de código ejecutable que establecen reglas inmutables.

De vuelta al contexto de las microredes, los contratos inteligentes pueden asumir el papel de un sistema de gestión de energía (EMS) centralizado, pero de manera totalmente distribuida. En los últimos años, se han investigado numerosos trabajos en este campo. Un ejemplo destacado de arquitectura para el comercio P2P, que emplea la tecnología blockchain, se encuentra en la Microred de Brooklyn [40]. Se trata de un mercado impulsado por órdenes, en el que los compradores y vendedores interactúan utilizando un libro de órdenes que asigna prioridad al postor más alto. En otro enfoque [72], se ha diseñado una plataforma de programación de energía con un día de anticipación basada en un contrato inteligente. Los participantes resuelven un problema de optimización distribuida mientras establecen coeficientes de intercambio bilaterales para designar preferencias de comercio. Además, en [76], se hace uso de la tecnología blockchain para maximizar la seguridad en el intercambio de energía entre microredes; y en [24], se propone una plataforma

de comercio de energía P2P descentralizada de dos capas, que consta de una capa de mercado para el comercio de energía entre agentes, y una capa de blockchain para almacenar los resultados en contratos inteligentes y llevar a cabo los acuerdos. Asimismo, en [37], se ha utilizado una blockchain de consorcio para diseñar un mercado de comercio de energía P2P híbrido, donde se negocia energía entre consumidores, prosumidores y la red principal, reduciendo las variaciones entre picos y promedios, así como el coste de la electricidad tanto para la red de servicios públicos como para los consumidores de energía. Este intercambio de energía se basa en la distancia física entre las partes, lo que contribuye a reducir las pérdidas en línea.

1.1 Objetivos

Este trabajo busca ahondar en la comprensión de la tecnología blockchain aplicada a microrredes o comunidades energéticas, haciendo especial hincapié en los conceptos subyacentes a dicha tecnología y en particular a la red de Ethereum que es la red más madura en términos de contratos inteligentes; y busca presentar y desarrollar los resultados expuestos en [67] en el ámbito de comunidades energéticas gestionadas por un algoritmo TBMPC y distribuido que es coordinado por un contrato inteligente. Además, la interacción entre los agentes y el contrato inteligente se gestiona a través de eventos emitidos por el contrato inteligente para informar sobre el logro de ciertos hitos de control y notificar la necesidad de llevar a cabo acciones adicionales. Para gestionar estos eventos, se ha utilizado la plataforma OpenZeppelin Defender, reconocida por su capacidad para supervisar eventos dentro de blockchain y automatizar la llamada de funciones de contratos inteligentes [7].

A diferencia de otros trabajos que han empleado enfoques TBMPC para lidiar con incertidumbres en contextos similares, como [74] o [31], este trabajo se centra en el aspecto distribuido y reemplazar la capa superior convencional, que generalmente involucra una autoridad central, por un contrato inteligente desplegado en Ethereum. Esta sustitución excluye la necesidad de terceros en la gestión de datos. En lo que respecta a la implementación distribuida, se ha empleado el algoritmo Feasible cooperation-based MPC (FC-MPC) [75]. En comparación con otros algoritmos de optimización distribuidos comúnmente utilizados en comunidades energéticas, como el Método lagrangiano aumentado (o más conocido como Alternating Direction Method of Multipliers, ADMM) en [72] o [42], el algoritmo FC-MPC presenta características ventajosas, como que todas las soluciones intermedias son factibles para el sistema en su conjunto, incluso si el algoritmo finaliza antes de alcanzar convergencia. A diferencia del ADMM, el algoritmo FC-MPC no parte de la suposición de que el problema de optimización es separable, lo que le otorga mayor robustez y aplicabilidad a una variedad más amplia de problemas.

La estructura del trabajo está organizada de la siguiente manera. El Capítulo 1 presenta una introducción a la temática planteada en el trabajo y los objetivos de éste; el Capítulo 2 centra su atención en describir la tecnología blockchain; el Capítulo 3 especifica la atención a la blockchain de Ethereum, centrándose en sus aspectos más importantes; el Capítulo 4 presenta de manera teórica algunos métodos de control de comunidades energéticas y el papel de blockchain en el proceso; el Capítulo 5 describe la plataforma de gestión energética diseñada; el Capítulo 6 muestra resultados numéricos de la plataforma previamente presentada; y el Capítulo 7 expone las conclusiones del trabajo.

2 Tecnología blockchain

En esta sección se hará un breve repaso a la historia de blockchain, los principales conceptos que la fundamentan y los principales sistemas operativos blockchain.

2.1 Historia de blockchain

La tecnología blockchain surgió en 2008 con la presentación del "white paper" de Bitcoin por parte de Satoshi Nakamoto, estableciendo así el marco inicial de una blockchain [47]. No obstante, muchos de los conceptos que sustentan esta tecnología tienen raíces que se remontan varios años atrás.

Los pilares fundamentales de la tecnología blockchain se establecen en el campo de la criptografía, con un enfoque claro en la seguridad de las comunicaciones y las transacciones digitales. Desde los albores de la década de 1970, destacados criptógrafos como Whitfield Diffie y Martin Hellman se dedicaron al desarrollo de protocolos de cifrado de clave pública [22], sentando así las bases necesarias para las aplicaciones futuras de la blockchain. No obstante, la concepción de una cadena de bloques distribuida todavía no había cobrado forma en ese momento.

La aparición del Manifiesto de los Cypherpunks en la década de 1990, un grupo de activistas comprometidos con la privacidad y la seguridad en línea, marcó un momento crítico. Figuras destacadas como Timothy C. May y Eric Hughes introdujeron la idea de que la privacidad constituye un derecho esencial y promovieron la construcción de sistemas de cifrado y anonimato. Esta corriente ideológica allanó el camino filosófico de la blockchain al abogar por la concepción de un sistema financiero y de comunicaciones descentralizado [35].

En 1991, los investigadores Haber y Stornetta introdujeron por primera vez el concepto de una "cadena de bloques criptográficamente asegurada" [30]. Esta innovadora solución permitió la concatenación de datos digitales con sellos temporales, asegurando su inmutabilidad e integridad. Un año más tarde, en 1992, se incorporaron los Árboles de Merkle al diseño, lo que mejoró la eficiencia de la blockchain al permitir la asociación de numerosos datos individuales con un único valor o "hash".

En 1998, Nicholas Szabo diseñó un mecanismo para una moneda digital descentralizada, que denominó "bit gold" [71]. Aunque este proyecto nunca se materializó, sentó las bases para desarrollos futuros en criptomonedas y contratos inteligentes. En 2004, Hal Finney introdujo el sistema "Reusable Proof of Work" (RPoW), que funcionaba como un prototipo de dinero virtual [26]. Este sistema resolvió el problema del "doble gasto" al registrar la propiedad de los tokens en un servidor de confianza, permitiendo a usuarios de todo el mundo verificar la precisión y la integridad de los datos en tiempo real.

El verdadero punto de inflexión se produjo en 2008 con la publicación del "white paper" de Bitcoin por parte de Satoshi Nakamoto [47]. En este documento, se propuso un sistema de "dinero electrónico puramente peer-to-peer" que permitiría intercambios económicos directos entre partes sin depender de instituciones financieras, dando lugar al nacimiento de la primera blockchain basada en la criptomoneda Bitcoin (BTC).

En 2014, la tecnología blockchain evolucionó más allá de su aplicación original en criptomonedas y dio lugar a lo que se conoce como Blockchain 2.0. Esta fase marcó el comienzo de un nuevo paradigma en el que la blockchain dejó de estar limitada al intercambio de criptomonedas y se abrió a una amplia gama de aplicaciones. Este avance se materializó gracias a la implementación de lenguajes de programación que permitieron la creación de "smart contracts" más complejos, como Solidity [10], que se utiliza en la blockchain Ethereum [15]. Estos contratos inteligentes habilitaron una amplia variedad de aplicaciones, desde la gestión de la cadena de suministro hasta la ejecución automática de acuerdos comerciales.

2.2 Definición de blockchain

Este trabajo no tiene como objetivo la descripción integral y exhaustiva de la tecnología blockchain ya que ésta revisión fue realizada en el trabajo anterior [68]. Sin embargo, se proporcionará un breve resumen a modo de facilitar la comprensión del presente trabajo.

Una blockchain es un tipo de estructura de datos digitales compartida, distribuida e inmutable que actúa como un registro público y seguro de transacciones digitales. Es fundamental entender los elementos clave que componen una blockchain para apreciar su funcionamiento integral. A continuación, proporcionaré una definición detallada de blockchain, enriquecida con los aspectos esenciales que la componen [21], [12], [48]:

2.2.1 Componentes de una blockchain

- **Bloques:** en esencia, una blockchain está conformada por una secuencia de bloques, donde cada bloque alberga un conjunto de transacciones digitales. Estas transacciones varían en su naturaleza, abarcando desde el intercambio de criptomonedas hasta la ejecución de contratos inteligentes o registros digitales diversos.
- **Cadenas:** estos bloques se organizan de manera secuencial en una cadena cronológica, lo que asegura la integridad de las transacciones. Cualquier modificación en un bloque anterior afectaría a todos los bloques posteriores, lo que confiere una elevada seguridad y resistencia a la manipulación.
- **Criptografía:** la seguridad en una blockchain se basa en técnicas criptográficas sólidas. Cada bloque contiene un hash criptográfico del bloque anterior y de todas las transacciones contenidas en el bloque actual. Esto crea una relación inmutable entre los bloques, protegiendo la integridad de los datos almacenados.
- **Red P2P (Peer-to-Peer):** las redes blockchain son inherentemente distribuidas, lo que significa que no dependen de un servidor central. En su lugar, múltiples nodos (ordenadores remotos) colaboran para verificar y registrar las transacciones. Cada nodo posee una copia completa de la blockchain en la que trabajan, lo que garantiza resistencia ante fallos y censura.
- **Consenso:** para añadir un nuevo bloque a la blockchain, la mayoría de los nodos en la red deben estar de acuerdo en que las transacciones son válidas. Esto se logra a través de algoritmos de consenso, como son la Prueba de Trabajo (PoW) o la Prueba de Participación (PoS). Estos algoritmos garantizan que solo las transacciones legítimas o no repetidas sean añadidas, evitando problemas como el de doble gasto (double spending problem) [19].

2.3 Funcionamiento de una Blockchain

Una vez repasados los elementos fundamentales que componen una blockchain, se proporcionará una descripción del proceso que es recorrido desde que un usuario intenta interactuar con una blockchain, hasta que dicha interacción se materializa en un conjunto de datos dentro de un nuevo bloque.

2.3.1 Interacción del usuario

1. **Solicitud de transacción:** el proceso comienza cuando un usuario decide realizar una transacción en la blockchain. Esto puede ser una transferencia de criptomonedas o la ejecución de un método de un contrato inteligente cualquiera, entre otras opciones.
2. **Creación de la transacción:** en el caso que se desee hacer una transacción monetaria, el usuario crea una transacción que incluye detalles como la dirección del destinatario, la cantidad a transferir, la firma digital y otros datos relevantes.
3. **Firma digital:** antes de que la transacción se envíe a la red, se lleva a cabo una firma digital. Esta firma implica el uso de la clave privada del remitente para garantizar la autenticidad de la transacción y la clave pública del destinatario para que este último pueda verificar su origen. La clave privada es necesaria para el proceso de cifrado, lo que asegura que solo el destinatario designado pueda descifrar y acceder a la información de la transacción. Este procedimiento se basa en un sistema criptográfico de clave pública, que garantiza la seguridad y la integridad de la comunicación.[65].

2.3.2 Transmisión de la Transacción

Una vez que la transacción se ha firmado digitalmente, se encuentra lista para ser transmitida a la red blockchain. Este proceso de transmisión se inicia mediante la conexión del nodo del remitente con otros nodos de la red, ya sea directamente o a través de un mecanismo de difusión de información. A continuación, se detallan los pasos clave involucrados en la transmisión de una transacción en una blockchain:

1. **Propagación a nodos vecinos:** el nodo remitente envía la transacción a los nodos vecinos en la red. Estos nodos vecinos pueden ser otros participantes en la red, que actúan como puntos de entrada para difundir la información. La transacción se propaga de nodo a nodo, creando un flujo de datos en toda la red.
2. **Validación inicial:** a medida que la transacción se propaga a través de los nodos, estos realizan una validación inicial para verificar su formato y estructura. Esto incluye comprobar si la firma digital es válida, si las claves públicas y privadas coinciden y si la transacción cumple con los requisitos específicos de la cadena de bloques.
3. **Difusión a la red completa:** la transacción se propaga de nodo a nodo hasta que alcanza la red completa de la blockchain. Esto garantiza que todos los nodos tengan conocimiento de la transacción pendiente y puedan considerarla para su inclusión en el siguiente bloque.
4. **Inclusión en el mempool:** después de la validación inicial, la transacción se agrega al "mempool" o "pool de transacciones pendientes" de cada nodo. El mempool es una especie de área de espera donde se almacenan temporalmente las transacciones no confirmadas. Aquí, las transacciones esperan a ser recogidas por los mineros y agregadas a un bloque.
5. **Espera para su inclusión en un bloque:** una vez que la transacción se encuentra en el mempool de la mayoría de los nodos, está lista para ser recogida por un minero. Los mineros seleccionan transacciones del mempool para incluirlas en un bloque que están a punto de crear.
6. **Confirmación de la transacción:** la transacción se agrega finalmente a un bloque cuando un minero resuelve con éxito un problema criptográfico y crea un nuevo bloque en la cadena. Una vez que el bloque se añade a la cadena, la transacción se considera confirmada y forma parte de la historia inmutable de la blockchain.

Es importante destacar que, durante este proceso, la transacción está criptográficamente asegurada y no puede ser alterada. Además, debido a que la blockchain es una red distribuida y descentralizada, múltiples copias de la cadena están disponibles en diferentes nodos, lo que garantiza la redundancia y la integridad de la información. La transacción, una vez confirmada, es visible públicamente en la blockchain y se convierte en parte de su historial permanente.

2.3.3 Mineros

Los mineros, introducidos en el punto anterior, son nodos especiales en la red blockchain responsables de agrupar transacciones en bloques y resolver complejos rompecabezas criptográficos como parte del proceso de consenso. Su objetivo es seleccionar un conjunto de transacciones de la mempool y los agrupan en un bloque. La selección se basa en varios factores, como tarifas de transacción y prioridad.

El "rompecabezas criptográfico" dependerá de cada blockchain, sin embargo, los más extendidos son la Prueba de Trabajo (Proof of Work, PoW), y la Prueba de Participación (Proof of Stake). En el caso que se utilice PoW, el primer minero capaz de resolver dicho problema es el que consigue validar y agregar el bloque a la cadena, además de recibir un incentivo económico [12].

2.3.4 Inclusión en la Blockchain

Una vez que el bloque es validado y agregado a la cadena, la transacción es considerada inmutable y forma parte del registro histórico de la blockchain. Esta nueva información se propaga por toda la red para que todos los nodos tengan una copia actualizada de la blockchain. Finalmente, la transacción se hace visible para cualquier usuario de la blockchain, de modo que tanto la transacción, como su inclusión en la cadena de bloques, puede ser verificada utilizando un exploradores de bloques.

2.4 Principales blockchain

En este capítulo, se exploran algunas de las principales blockchain que han dado forma al panorama de esta tecnología revolucionaria. Cada una de estas blockchain tiene sus propias características y casos de uso distintivos, lo que las convierte en pilares de diversas industrias. Profundizaremos en las tecnologías subyacentes, las innovaciones clave y los casos de uso más destacados.

2.4.1 Bitcoin

Bitcoin, lanzada en enero de 2009 por el seudónimo Satoshi Nakamoto, marcó el inicio de la revolución de las criptomonedas y las blockchains. Esta red se basa en un libro de contabilidad público descentralizado conocido como blockchain. A continuación, se describen los aspectos clave de la red Bitcoin:

- **Arquitectura descentralizada:** la red Bitcoin se basa en un modelo descentralizado, lo que significa que no está controlada por ninguna entidad central, como un banco o un gobierno. En cambio, está mantenida por una red de nodos distribuidos en todo el mundo. Estos nodos son ordenadores que almacenan una copia completa de la blockchain de Bitcoin y trabajan juntos para verificar y registrar transacciones.
- **Consenso y minería:** la red Bitcoin utiliza el algoritmo de consenso PoW. En PoW, los nodos de la red, conocidos como mineros, compiten para resolver un rompecabezas criptográfico complejo. Este rompecabezas es un problema matemático que requiere una gran cantidad de poder computacional para ser resuelto. El funcionamiento de PoW es el siguiente:
 1. **Transacciones pendientes:** cuando un usuario realiza una transacción de Bitcoin, esta se agrega a un grupo de transacciones pendientes conocido como "mempool."
 2. **Selección de transacciones:** los mineros seleccionan un grupo de transacciones de la mempool y las incluyen en un bloque.
 3. **Prueba de trabajo:** el objetivo de los mineros es encontrar un número llamado "nonce" que, cuando se combina con las transacciones y otros datos del bloque, genere un valor hash [69] (una cadena de caracteres) que cumpla con ciertos requisitos. En otras palabras, deben encontrar un nonce que haga que el hash del bloque comience con un cierto número de ceros.
 4. **Competencia minera:** los mineros compiten para encontrar el nonce correcto. Esto implica realizar numerosos intentos y consumir una gran cantidad de energía eléctrica y recursos computacionales.
 5. **Primer minero en encontrar el nonce:** el primer minero en encontrar el nonce correcto difunde su bloque a la red para su verificación.
 6. **Verificación de la red:** los otros nodos de la red verifican si el nuevo bloque cumple con las reglas de consenso, lo que incluye la validación de las transacciones y la confirmación de que se ha encontrado un nonce válido.
 7. **Recompensas y confirmación:** si el bloque es validado, el minero recibe una recompensa en Bitcoin [47] (actualmente, 6.25 BTC más las tarifas de transacción) y el bloque se agrega a la cadena de bloques de Bitcoin. Este proceso se conoce como "minería."
- **Limitación de suministro:** una característica distintiva de Bitcoin es su límite de suministro de 21 millones de monedas. Esto significa que no se pueden crear más Bitcoins una vez que se alcance este límite. Este límite en el suministro se diseñó para combatir la inflación y preservar el valor a lo largo del tiempo.
- **Seguridad:** la red Bitcoin utiliza criptografía para garantizar la seguridad de las transacciones. Cada usuario tiene un par de claves: una clave pública (que actúa como una dirección de recepción) y una clave privada (necesaria para firmar transacciones). La seguridad de Bitcoin se ha mantenido sólida durante años, lo que ha contribuido a su aceptación global como una forma segura de almacenar y transferir valor.
- **Uso y almacenamiento:** Bitcoin se utiliza principalmente como una reserva de valor y un medio de intercambio. Los usuarios pueden comprar, vender e intercambiar Bitcoins en mercados en línea y utilizarlos para comprar bienes y servicios en un número creciente de empresas. Además, muchas personas almacenan sus Bitcoins en billeteras digitales o hardware wallets para mantenerlos seguros.

La red Bitcoin, con su innovador algoritmo PoW, es la piedra angular del ecosistema blockchain y ha inspirado la creación de numerosas otras criptomonedas y blockchains. La combinación de seguridad y descentralización hace que Bitcoin sea un fenómeno disruptivo en el mundo financiero.

2.4.2 Ethereum

Ethereum [15], lanzada en 2015 por el programador Vitalik Buterin, es una plataforma blockchain que se destaca por su capacidad para ejecutar contratos inteligentes, que son programas informáticos que se ejecutan automáticamente cuando se cumplen ciertas condiciones. A continuación, se describen aspectos básicos de la red Ethereum:

- Algoritmo de consenso: a diferencia de Bitcoin, que utiliza el algoritmo PoW, Ethereum ha estado en proceso de transición hacia PoS (Prueba de Participación) con la actualización Ethereum 2.0 [55]. Con PoS, los nodos que desean validar transacciones deben bloquear una cierta cantidad de Ether (la criptomoneda nativa de Ethereum) como garantía. La seguridad de la red se basa en la cantidad de Ether en juego y no en la capacidad de cálculo, como en PoW.
- Contratos inteligentes: Ethereum es conocida principalmente por su capacidad para ejecutar contratos inteligentes. Estos contratos son programas escritos en lenguajes de programación como Solidity [10] y se almacenan en la blockchain de Ethereum. Gracias a ellos, las posibilidades de la blockchain de Ethereum trascienden al mero intercambio monetario, permitiendo el desarrollo de aplicaciones descentralizadas.
- Aplicaciones descentralizadas: gracias a los smart contracts, Ethereum permite el desarrollo de aplicaciones descentralizadas, que son aplicaciones que se ejecutan en la red Ethereum en lugar de en servidores centralizados. Estas aplicaciones pueden abarcar desde juegos y finanzas descentralizadas hasta sistemas de votación.

Sin lugar a dudas, lo más interesante de la red de Ethereum es la posibilidad de programar contratos inteligentes, por lo que esta red será descrita con mayor detenimiento en el capítulo 3.

2.4.3 Polkadot

Polkadot es una plataforma que se centra en la interoperabilidad entre blockchains y en abordar los problemas de escalabilidad que presentan múltiples blockchains. Esta capacidad permite la transferencia de datos entre blockchain públicas, abiertas y sin permisos, así como en blockchains privadas y con permisos. Esto posibilita la creación de aplicaciones que pueden acceder a datos que requieren permisos de una blockchain privada y utilizarlos en una blockchain pública [63]. A continuación, se describen sus características clave:

- Algoritmo de Consenso [77]: utiliza un mecanismo de consenso llamado Nominated Proof of Stake (NPoS), que combina PoS con una capa de nominadores. Los titulares de tokens DOT (la criptomoneda de Polkadot) pueden nominar a validadores y recibir recompensas por hacerlo. NPoS se utiliza para seleccionar a los validadores que aseguran la red y validan transacciones.
- Subcadenas (Parachains): introduce el concepto de parachains que pueden ser blockchains independientes o cadenas laterales (sidechains) conectadas a la blockchain principal de Polkadot. Esto permite una mayor escalabilidad y flexibilidad al permitir que múltiples blockchains coexistan y se comuniquen.
- Seguridad y Escalabilidad: Polkadot está diseñada para priorizar la seguridad y la escalabilidad al permitir que las parachains compartan la seguridad de la cadena principal. Esto significa que las subcadenas no tienen que preocuparse por su propia seguridad y pueden centrarse en sus propias características y aplicaciones.
- Usos Principales: se utiliza para construir una red de cadenas interconectadas y seguras. Permite a los desarrolladores crear aplicaciones descentralizadas y blockchains especializadas que pueden comunicarse entre sí. Polkadot es especialmente relevante en el desarrollo de soluciones empresariales y gubernamentales que requieran de alta seguridad y escalabilidad.

2.4.4 Chainlink

Chainlink (LINK) es una plataforma que se distingue por su papel crucial en la interconexión entre blockchains y fuentes de datos del mundo real. Para ello, Chainlink introduce el concepto de oráculos. Los oráculos son intermediarios que permiten que los contratos inteligentes en las blockchains accedan y utilicen información externa de manera confiable y segura. A continuación, se describen las características principales de Chainlink:

- Oráculos descentralizados: Chainlink opera una red de oráculos descentralizados que actúan como intermediarios de confianza entre las blockchains y las fuentes de datos del mundo real. Estos oráculos están compuestos por nodos de Chainlink, cada uno de los cuales puede proporcionar datos y servicios a las aplicaciones de blockchain.
- Seguridad y fiabilidad: la red de Chainlink está diseñada para garantizar la integridad de los datos proporcionados por los oráculos. Utiliza contratos inteligentes específicos llamados "contratos oráculo" para verificar y autenticar los datos antes de que se utilicen en los contratos inteligentes de las aplicaciones.
- ¿Algoritmo de consenso? En lugar de un algoritmo de consenso, Chainlink se basa en la reputación y la competencia de los nodos oráculos para garantizar la integridad de los datos proporcionados. Los desarrolladores de aplicaciones pueden seleccionar nodos oráculos basados en su historial de rendimiento y confiabilidad.
- Token LINK: el token nativo de Chainlink, llamado LINK, se utiliza para compensar a los nodos oráculo por sus servicios. También se utiliza como garantía para asegurar la integridad de los datos proporcionados por los oráculos. Los titulares de LINK pueden participar en la red como nodos o utilizar el token en aplicaciones basadas en Chainlink.

3 Ethereum

La blockchain de Ethereum cobra una especial importancia en el desarrollo de este trabajo puesto que es el entorno más maduro que permite el desarrollo y despliegue de contratos inteligentes. El objetivo último es el de presentar una plataforma de gestión energética distribuida basada en blockchain, sin embargo, primero se dará una profunda visión de Ethereum empezando por su recorrido histórico, sus componentes, el "estado de Ethereum", la Ethereum Virtual Machine (EVM), y los desafíos y futuro de Ethereum.

3.1 Historia de Ethereum

En la Sección 1, se abordó la historia general de la tecnología blockchain. Sin embargo, ahora profundizaremos en la historia específica de Ethereum [52], ampliamente reconocido como el segundo proyecto más relevante en el panorama blockchain actual, únicamente superado por Bitcoin. Este reconocimiento se refleja de manera contundente en su capitalización de mercado actual, la cual, al 6 de noviembre de 2023, asciende a casi 227.000 millones de euros [18]. Ethereum ha demostrado de manera innegable su importancia y liderazgo en el ecosistema blockchain.

En el año 2013, Vitalik Buterin formuló la visión de Ethereum a través del *white paper* de Ethereum [15], un documento que proponía la creación de una plataforma revolucionaria que permitiría a los desarrolladores crear aplicaciones descentralizadas (DApps) con contratos inteligentes en una blockchain. La idea principal era superar las limitaciones de Bitcoin y proporcionar un lenguaje de programación versátil que facilitara el desarrollo blockchain.

Fue dos años más tarde, en julio de 2015, cuando Ethereum fue lanzado oficialmente. La plataforma introdujo la noción de contratos inteligentes, que revolucionó la forma en que las transacciones se llevaban a cabo. Los contratos inteligentes permiten la ejecución automática de acuerdos siempre y cuando se cumplan ciertas condiciones. Este evento impulsó una ola de innovación inédita en el espacio blockchain.

En el año 2016 tuvo un evento crucial que cambió la historia de Ethereum, cuando la red experimentó una bifurcación (ampliamente conocido como *hard fork*). Esto ocurrió después de un incidente de seguridad en la organización autónoma descentralizada (DAO) de Ethereum, que resultó en el robo de una gran cantidad de ether (el concepto de DAO será explicado a lo largo de esta sección). La comunidad se dividió en cuanto a cómo abordar este problema. Algunos creían que Ethereum debería permanecer inmutable y no intervenir en el código de contrato inteligente, argumentando que hacerlo sería contraproducente para la inmutabilidad de la cadena de bloques. Otros argumentaron que, dado que Ethereum se consideraba una plataforma flexible, debería abordar el problema y devolver los fondos a los inversores afectados.

Finalmente, después de una serie de debates y votaciones, la mayoría de la comunidad de Ethereum optó por realizar un duro fork en la cadena de bloques. Esto implicó la creación de una nueva cadena, la que se conoce a día de hoy como Ethereum, que contenía la transacción que revirtió las acciones del atacante y devolvió los fondos a los inversores de la DAO. La blockchain original se mantuvo y se convirtió en lo que hoy se conoce como *Ethereum Classic* [60], que optó por mantener la inmutabilidad.

Después del incidente, Ethereum introdujo nuevas propuestas de gobernanza y actualizaciones en su blockchain. Por ejemplo, el Hard Fork de Byzantium en 2017 introdujo nuevas tecnologías que facilitaron la construcción de blockchains adicionales sobre Ethereum. Estas blockchains, conocidas como Layer 2 (L2) blockchains (como Polygon [36]) se han convertido en una parte cada vez más significativa del ecosistema de Ethereum y que buscan solucionar los problemas de escalabilidad.

Ethereum Daily Transactions Chart

Source: Etherscan.io

Click and drag in the plot area to zoom in

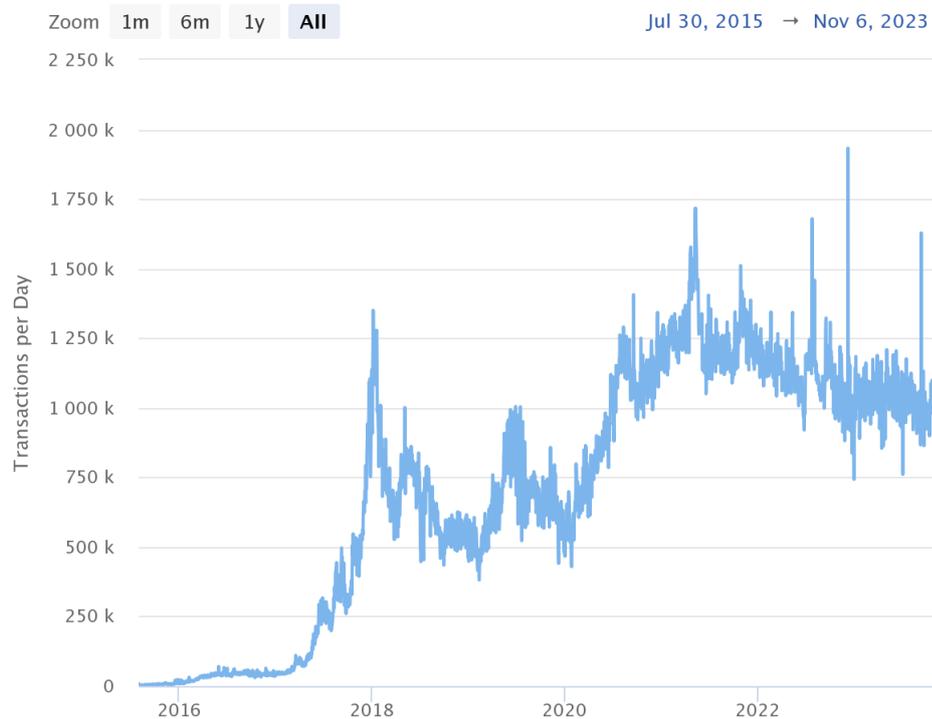


Figura 3.1 Transacciones realizadas en Ethereum entre 2015 y 2023 [25].

Las actualizaciones Istanbul y Constantinople en 2019 se centraron en optimizar la estructura de tarifas de gas de Ethereum e introducir soluciones adicionales de escalabilidad. Además de lanzar nuevas soluciones para L2 blockchains, la actualización de Constantinople allanó el camino para una transición fluida al mecanismo de consenso de prueba de participación (proof of stake, PoS).

A diferencia de PoW, PoS requiere que los validadores bloqueen una criptomoneda nativa en la blockchain. Cualquiera que bloquee una cantidad determinada de ether puede confirmar nuevos bloques y reclamar recompensas. La transición del esquema PoW de Ethereum a un esquema PoS beneficiaría la blockchain en dos aspectos fundamentales: mejoras en escalabilidad y en la reducción de la huella de carbono.

A finales de 2020, Ethereum introdujo una blockchain de PoS llamada *Beacon Chain*, que necesitaba 16.384 depósitos de 32 ETH para empezar a funcionar. Finalmente, la Beacon Chain reemplazó la blockchain de PoW de Ethereum en un evento llamado *The Merge* en 2022. Antes de esta transición, cualquier persona con 32 ETH podía bloquearlos en la Beacon Chain para comenzar a ganar recompensas por participar en la validación y seguridad de la red. El efecto inmediato más notable de *The Merge* fue la reducción de la huella de carbono de Ethereum debido a que ya no eran necesarios mineros que anteriormente utilizaban múltiples tarjetas gráficas. Esto se tradujo en una reducción del consumo de electricidad y emisiones netas estimadas en un 99.95%. Gracias al consenso de PoS de Ethereum, pasó de ser uno de los mayores contaminadores a convertirse en una blockchain respetuosa con el medio ambiente. Además, se estimó que, después del paso a PoS, la creación de nuevos ETH diarios en concepto de recompensas para el mantenimiento de la red pasaría de 13.000 a 1.700 ETH diarios [56]. En la Figura 3.1 se puede observar el número de transacciones que ha experimentado la blockchain de Ethereum desde julio de 2015 hasta noviembre de 2023. Aunque pueda parecer que está experimentando un retroceso en el número de transacciones desde el año 2021, es necesario tener en cuenta el panorama macroeconómico, dado que años de bonanza económica en el sector de las criptomonedas se traduce en un incremento en las transacciones y, viceversa. Sin embargo, a pesar de haber experimentado una fuerte bajada en valor de mercado desde el *all time high* marcado en 2021 con aproximadamente un valor de 4.700 \$ por moneda, el ratio de caída del número de transacciones ha sido muy inferior al de su precio: signo del desarrollo que se ha, y sigue haciendo durante estos años.

3.2 Organización Autónoma Descentralizada (DAO)

Antes de desarrollar los componentes principales de la red de Ethereum, se explicará el concepto de DAO introducido en la Sección 3.1, que es un concepto esencial e inherente al ecosistema de Ethereum [66].

3.2.1 Orígenes de las DAOs

Las DAOs son un concepto innovador que busca codificar las reglas y procesos de toma de decisiones de una organización en forma de contratos inteligentes en la blockchain de Ethereum. La idea detrás de las DAO es eliminar la necesidad de documentos y personas en la administración de una organización, creando una estructura con un control descentralizado.

La primera DAO, en su sentido más amplio, es el propio Bitcoin. La gobernanza de Bitcoin se basa en un consenso entre su equipo principal y su red de mineros. Sin embargo, las DAO más conocidas y pioneras se construyeron en la plataforma Ethereum.

3.2.2 Funcionamiento de una DAO

El funcionamiento de una DAO implica los siguientes pasos:

1. Creación de contratos inteligentes: un grupo de personas diseña y codifica los contratos inteligentes que definirán las reglas y operaciones de la DAO. Estos contratos se ejecutarán automáticamente en cada nodo de la red Ethereum.
2. Periodo de financiación inicial: para que la DAO tenga los recursos necesarios para operar, se inicia un período de financiamiento en el que las personas pueden contribuir fondos comprando tokens que representan la propiedad o participación en la DAO. Este proceso es conocido como *crowdsale* o *initial coin offering* (ICO)."
3. Operación de la DAO: una vez finalizado el periodo de financiamiento, la DAO comienza a funcionar. Los participantes que han adquirido tokens pueden presentar propuestas sobre cómo gastar los fondos de la DAO. Estas propuestas pueden estar relacionadas con proyectos, inversiones o cualquier otro aspecto de la organización.
4. Votación y toma de decisiones: los titulares de tokens de la DAO tienen derecho a votar en las propuestas presentadas. La toma de decisiones se basa en la cantidad de tokens que poseen los participantes, lo que se conoce como gobernanza basada en tokens. Esto permite que los titulares de tokens influyan en la dirección y las acciones de la DAO.
5. Ejecución de propuestas: una vez que una propuesta es aprobada por la mayoría de los votantes, se ejecuta automáticamente a través de los contratos inteligentes de la DAO. Esto garantiza que las decisiones se implementen sin necesidad de intermediarios.

Es importante destacar que los tokens de las DAO no representan propiedad de la organización, sino más bien contribuciones que otorgan derechos de voto. En la mayoría de los casos, las DAO operan sin una entidad o individuo propietario, ya que son simplemente programas de software que se ejecutan en la red Ethereum.

3.2.3 El caso de "The DAO"

Uno de los casos más notorios relacionados con las DAO es el de "The DAO," que lanzada el 30 de abril de 2016, con un período de financiación de 28 días. Logró recaudar aproximadamente una cantidad de ETH equivalente a 150 millones de dólares con la participación de más de 11.000 inversores [70].

No obstante, a pesar del éxito alcanzado, surgieron inquietudes entre expertos en informática debido a la identificación de vulnerabilidades en el código que planteaban posibles riesgos para la seguridad de los fondos. Específicamente, Stephan Tual, uno de los creadores de "The DAO," anunció el 12 de junio que se había detectado un "error de llamada recursiva" en el software, aunque enfatizó que "los fondos de la DAO no estaban en peligro." Esta situación desencadenó una serie de eventos que culminaron en uno de los incidentes más notables en la historia temprana de Ethereum, como se exploró previamente en la Sección 3.1.

3.3 Componentes de Ethereum

La red de Ethereum es un sistema complejo que consta de varios componentes esenciales que trabajan en conjunto para habilitar su funcionalidad. Estos componentes son fundamentales para comprender cómo opera Ethereum. A continuación, se describen los componentes clave de la red de Ethereum:

- **Ether (ETH):** Ether es la criptomoneda nativa de Ethereum. Similar a Bitcoin, se utiliza como medio de intercambio y almacén de valor. Los usuarios pueden adquirir Ether y utilizarlo para pagar transacciones en la red de Ethereum. También se requiere Ether para pagar las tarifas de gas que mantienen la red y compensan a los mineros y validadores.
- **Smart Contracts:** los smart contracts, o contratos inteligentes, son programas informáticos que se ejecutan en la red de Ethereum. Estos contratos están diseñados para automatizar y ejecutar acuerdos y transacciones sin necesidad de intermediarios. Los smart contracts se ejecutan en la Ethereum Virtual Machine (EVM) y son fundamentales para una variedad de aplicaciones descentralizadas. El lenguaje de programación utilizado para la codificación de los smart contracts es Solidity [10].
- **Solidity:** es un lenguaje de programación Turing Completo desarrollado para escribir smart contracts en la plataforma Ethereum y ser utilizado sobre la EVM.
- **Nodos:** en la red de Ethereum, los nodos son los ordenadores o unidades de cálculo que forman parte de la infraestructura. Estos nodos pueden ser de dos tipos: completos o ligeros. Los nodos completos almacenan toda la historia de la blockchain de Ethereum y validan todas las transacciones. Por otro lado, los nodos ligeros solo almacenan las cabeceras de los bloques y no validan todas las transacciones. Los nodos son esenciales para mantener la integridad de la red, ya que participan en la validación de las transacciones y la ejecución de contratos inteligentes.
- **Direcciones:** las direcciones en Ethereum son como las cuentas bancarias en el mundo real. Cada dirección es única y se utiliza para enviar y recibir ether (ETH) o ejecutar contratos inteligentes. Las direcciones están basadas en criptografía de clave pública y clave privada, lo que significa que una dirección se deriva de una clave pública y se protege con una clave privada. La clave privada es necesaria para firmar transacciones y demostrar la propiedad de los activos en esa dirección.
- **Transacciones:** las transacciones son la forma en que se envían fondos o se interactúa con contratos inteligentes en la red de Ethereum. Cada transacción incluye información como la dirección del remitente, la dirección del destinatario, la cantidad de ether involucrada y datos adicionales si la transacción está interactuando con un contrato inteligente. Las transacciones deben ser firmadas digitalmente con la clave privada del remitente para ser válidas. Una vez que se envía una transacción a la red, los nodos trabajan para validarla y agregarla a un bloque.
- **Bloques:** los bloques son unidades de datos que contienen múltiples transacciones. Estos bloques son la forma en que se registran las transacciones en la blockchain de Ethereum. Cada bloque contiene un encabezado con información relevante, como el número de bloque, la marca de tiempo y un enlace al bloque anterior. Los bloques son fundamentales para la seguridad y la integridad de la red, ya que garantizan que todas las transacciones sean ordenadas y resistentes a la manipulación.
- **Actualizaciones de la red:** la red de Ethereum se actualiza regularmente para mejorar su rendimiento y seguridad. Estas actualizaciones a menudo se realizan a través de hard forks y pueden introducir mejoras técnicas y cambios en las reglas de consenso.

Una vez definidos los principales actores en Ethereum, se presenta el concepto de **Ethereum Virtual Machine (EVM)** [4]. Tal y como se define en su documentación, el lugar en el que reside físicamente la EVM no puede ser descrito de la misma manera que no se señala un punto de una nube o una ola del mar, sino que existe como una única entidad mantenida por miles de ordenadores ejecutando el cliente de Ethereum.

El protocolo de Ethereum existe únicamente con el objetivo de garantizar la operación continua, ininterrumpida e inmutable de la EVM. Es el entorno en el que residen todas las cuentas y contratos inteligentes de Ethereum. En cualquier bloque de Ethereum, desde el primero hasta el último, Ethereum tiene un único "estado canónico", y la EVM es la que establece las reglas para calcular un nuevo estado válido de bloque a bloque.

Con frecuencia, se emplea la metáfora de un "registro distribuido" para explicar blockchains como Bitcoin, que posibilitan una existencia de una moneda descentralizada mediante herramientas criptográficas. Este registro tiene un registro de la actividad que está sujeto a cumplir un conjunto de normas que regulan lo

que se puede o no se puede hacer para alterar el registro. Por ejemplo, en Bitcoin, una dirección no puede gastar más monedas de las que ha recibido previamente. Estas reglas fundamentales son la base de todas las transacciones en Bitcoin y numerosas otras blockchains.

La criptomoneda nativa de Ethereum, el Ether, sigue prácticamente las mismas reglas intuitivas, sin embargo, también presenta una funcionalidad mucho más poderosa y ya presentada: los contratos inteligentes. De modo que para describir este concepto se requiere una analogía más sofisticada. En lugar de un "registro distribuido", Ethereum se asemeja a una máquina de estados distribuida. El estado de Ethereum es una estructura de datos amplia que no solo almacena todas las cuentas y sus saldos, sino que es una máquina de estados que puede cambiar entre bloques conforme se vayan ejecutando los conjuntos de reglas predefinidas, y que ejecutan código máquina arbitrario.

El estado de Ethereum puede ser descrito a través de la siguiente función de transición:

$$Y(S_i, T) = S_{i++},$$

donde S_i hace referencia a un estado válido anterior, T es el conjunto de nuevas transiciones, e $Y(S, T)$ es la función de transición de Ethereum que produce el nuevo estado S_{i++} . El estado, en Ethereum, es una estructura gigante de datos que recibe el nombre de *Merkle Patricia Trie* [57], que mantiene a todas las cuentas unidas a través de hashes y con la capacidad de ser reducidas a una única raíz de hash guardada en la blockchain.

3.4 Principales usos de Ethereum

Algunas de los casos de uso más importantes y destacados de Ethereum durante los últimos años son los siguientes:

- Finanzas descentralizadas: Ethereum es la principal plataforma en la que se han construido aplicaciones de Finanzas Descentralizadas. Esto incluye préstamos, intercambio de activos, comercio de derivados, y muchas otras actividades financieras que, históricamente, siempre se han realizado a través de instituciones financieras centralizadas. Ethereum permite a los usuarios acceder y administrar sus finanzas sin necesidad de intermediarios.
- Tokens no fungibles (NFTs): los NFTs se han convertido en una forma de propiedad digital y autenticación única, impulsando un mercado de arte digital en constante crecimiento.
- Juegos y entretenimiento: durante los últimos años el sector de los videojuegos se ha interesado en el desarrollo y despliegue de juegos en línea y entretenimiento digital. Los juegos basados en blockchain permiten la propiedad y el comercio de elementos de juego, lo que atrajo a jugadores y coleccionistas.
- Supply chain y logística: donde han nacido aplicaciones basadas en rastrear el movimiento de productos y activos a lo largo de largas cadenas de suministro, otorgando una mayor transparencia y confianza en dichas cadenas.
- Atención médica y salud: se están dedicando esfuerzos en desarrollar aplicaciones de atención médica en Ethereum para gestionar registros médicos, facilitar la interoperabilidad entre sistemas de atención médica, a la vez que se garantiza la seguridad de los datos de los pacientes.

3.5 El trilema de la escalabilidad descentralizada

El trilema de aumentar la escalabilidad en un sistema descentralizado se muestra en la Figura 3.2, donde se hace referencia a uno de los mayores desafíos a los que se enfrentan redes blockchain, como Ethereum. Consiste en tres elementos que se consideran fundamentales como son la seguridad, la descentralización y la escalabilidad. Sin embargo, el trilema consiste en que es muy difícil la optimización simultánea de los tres aspectos, y no sólo eso, sino que el aumento de uno de ellos, suele llevar asociado un impacto negativo en uno de los otros dos. A continuación, se describen los tres componentes del desafío:

1. Seguridad: en el contexto de las blockchains, la seguridad se refiere a la resistencia de la red contra ataques maliciosos y a la integridad de las transacciones. Cuanto más segura sea una red, menos vulnerable será a amenazas, como ataques del 51 % o ataques de doble gasto [1]. Una alta seguridad es esencial para que las blockchains sean confiables y cumplan su función.

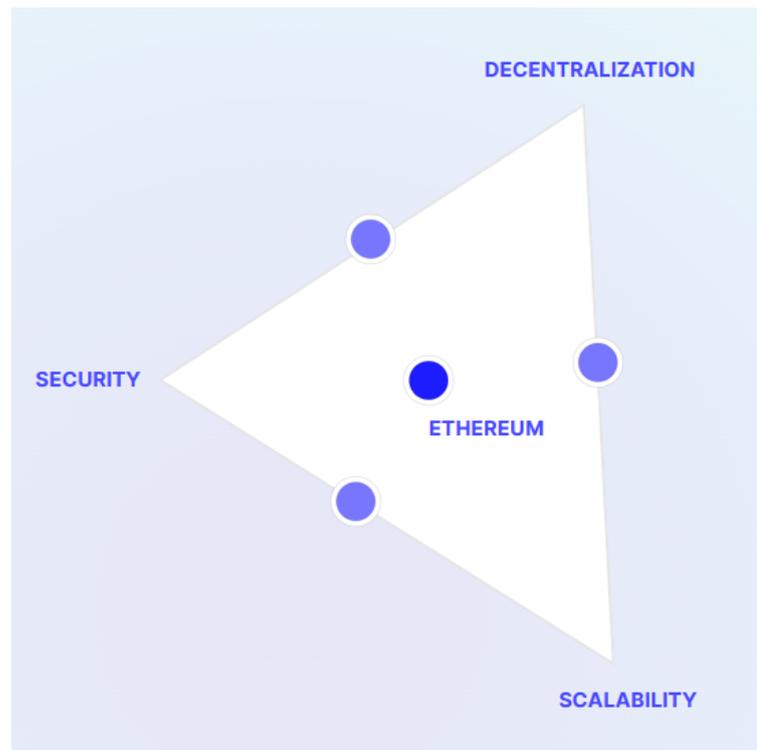


Figura 3.2 Ethereum scalability trilemma [50].

2. **Descentralización:** la descentralización se refiere a la distribución de nodos en la red y a la capacidad de cualquier persona para unirse a la red como nodo validador. Cuanto más descentralizada sea una red, menos control ejercerá un grupo selecto de participantes o entidades, lo que la hace más resistente a la censura y la manipulación.
3. **Escalabilidad:** la escalabilidad se refiere a la capacidad de la red para manejar un gran número de transacciones o actividades sin ralentizarse ni congestionarse. Una blockchain escalable puede procesar un alto volumen de transacciones de manera eficiente y, por lo tanto, es adecuada para casos de uso de gran demanda.

El trilema plantea que mejorar uno de estos aspectos puede hacer que otro se deteriore. Por ejemplo, si se busca aumentar la escalabilidad mediante el aumento de la velocidad de las transacciones, es posible que se deba hacerlo a expensas de la descentralización o la seguridad. Esto se debe a que, a menudo, lograr una mayor escalabilidad implica concesiones, como permitir que un menor número de nodos (lo que disminuye la descentralización) o reducir los mecanismos de seguridad para acelerar el procesamiento de transacciones.

La resolución efectiva del trilema de la escalabilidad es uno de los desafíos más importantes que enfrentan los desarrolladores de blockchain. En los últimos años, se han propuesto diversas soluciones, como las redes de capa 2, la transición a protocolos de consenso de Prueba de Participación (PoS) y la implementación de sharding, para abordar este desafío y encontrar el equilibrio adecuado entre seguridad, descentralización y escalabilidad en las blockchains.

3.5.1 Una visión un poco más detallada dentro de la escalabilidad

Con el crecimiento de usuarios en Ethereum, la blockchain ha enfrentado limitaciones de capacidad, aumentando los costes de la red y generando la necesidad de "soluciones de escalabilidad". Se están investigando, probando e implementando múltiples soluciones que abordan estas limitaciones desde enfoques diversos. El objetivo principal de la escalabilidad es aumentar la velocidad de las transacciones y el número de transacciones por segundo, sin sacrificar la descentralización o la seguridad, tal y como se vio en la Sección 3.5.

Conceptualmente, la escalabilidad se puede abordar desde dos puntos de vista: escalabilidad en cadena (on-chain) o escalabilidad fuera de cadena (off-chain).

Escalabilidad on-chain: Sharding

Sharding es el proceso de dividir una base de datos. Una vez dividida, cada conjunto en el que se ha dividido la base de datos total sería asignado a un conjunto de validadores. Estos validadores sólo tendrían que prestar atención a esta fracción de la red, de modo que no habría que prestar atención de todo Ethereum. Este proceso era uno de los objetivos principales de Ethereum, y así constaba en la hoja de ruta del proyecto. Sin embargo, el rápido desarrollo de las rollups de capa 2 (serán explicadas a continuación) han llevado a la comunidad de Ethereum a favorecer el escalado basado en rollups en lugar de mediante sharding. Para más información, consultar las siguientes referencias [54], [51], [53].

Escalabilidad off-chain

Las soluciones off-chain significa que se implementan por separado del protocolo de Ehtereum. Algunas soluciones, conocidas como soluciones de capa 2, derivan su seguridad directamente del consenso de Ethereum en la capa 1, como las optimistic rollups o zero-knowledge rollups. Las rollups ejecutan transacciones fuera de la cadena y envían los datos a la capa 1 para el consenso. Dentro de las rollups existen dos tipos:

- Optimistic rollups: asume la validez de las transacciones por defecto y solo realiza la verificación completa cuando aparece un desafío [58].
- Zero-knowledge rollups: realiza el cálculo fuera de la cadena y presenta una prueba de validez completa a la cadena [59].

4 Gestión energética y blockchain

En este capítulo se presentará con detenimiento el concepto de comunidades energéticas, los desafíos que presentan, y la propuesta de una plataforma de gestión energética distribuida para controlarlas basada en la tecnología blockchain, y más particularmente en los contratos inteligentes de la red Ethereum.

4.1 Comunidades energéticas

Las comunidades energéticas, en la actualidad, se han convertido en un tema de creciente interés en el contexto de la gestión de recursos energéticos. Estas comunidades son agrupaciones locales que buscan gestionar sus propios recursos de manera eficiente [13]. En esencia, una comunidad energética es un grupo de individuos, hogares, empresas u organizaciones que se asocian para generar, compartir y consumir energía de manera colaborativa. Estas comunidades pueden aprovechar una variedad de fuentes de energía, como paneles solares, turbinas eólicas o sistemas de almacenamiento de energía, para satisfacer sus necesidades locales. La energía producida se comparte dentro de la comunidad, lo que a menudo implica la venta o intercambio de excedentes de energía renovable.

Este enfoque ofrece una serie de ventajas. En primer lugar, empodera a las comunidades para tomar decisiones sobre su propia infraestructura energética, lo que a menudo conduce a un mayor compromiso y responsabilidad en la gestión de recursos energéticos. Esto tiene consecuencias directas en términos de costes energéticos e independencia de la comunidad con respecto a la red externa, lo que aumenta la resiliencia ante posibles interrupciones en el suministro.

Sin embargo, la implementación efectiva de comunidades energéticas presenta numerosos desafíos, desde técnicos hasta regulatorios o políticos:

- Regulaciones y políticas: las regulaciones y políticas energéticas existentes a menudo no están diseñadas para acomodar estructuras de comunidades energéticas. Esto puede dificultar la adopción y el funcionamiento de estos proyectos.
- Inversión inicial: la instalación de infraestructura renovable, como paneles solares o turbinas eólicas, puede requerir una inversión inicial significativa, lo que puede ser un obstáculo para algunas comunidades. Sin embargo, superada esta barrera, el *return on investment* (ROI) se aproxima a los 5 años [27].
- Tecnología y gestión: la implementación de tecnología adecuada y algoritmos de control son aspecto fundamental y que debe ser elegido cuidadosamente.
- Participación comunitaria: el éxito de una comunidad energética depende en gran medida de la participación y colaboración activa de los miembros de la comunidad.
- Integración en la red: la conexión de una comunidad energética a la red eléctrica existente y la gestión de la energía generada y consumida pueden requerir soluciones técnicas y reglamentarias.

4.2 Métodos

La gestión de microrredes o comunidades energéticas es un tema sujeto a amplia investigación durante los últimos años. Entre las técnicas de control que se han utilizado, destaca el empleo del Control Predictivo

basado en Modelo (Model Predictive Control, MPC), que destaca por ser una estrategia de control avanzada y versátil. Esta versatilidad hace que el empleo de controladores MPC sea demandado a la hora de gestionar microrredes o comunidades energéticas debido a su constante evolución, además de su adaptabilidad para lidiar con las complejidades asociadas a la inclusión de numerosos recursos energéticos distribuidos (como paneles solares, vehículos eléctricos, baterías, etc.) que han ido llegando durante los últimos años a los hogares.

MPC es un enfoque de control que se basa en la construcción y el uso de modelos matemáticos del sistema que se está controlando. Se utilizan estos modelos para predecir la evolución futura del sistema y determinar una secuencia óptima de acciones de control. Una de las singularidades del MPC radica en que, a pesar de que se calcula una secuencia completa de acciones de control óptimas, sólo la primera es implementada en el proceso mientras que las restantes son descartadas. En el siguiente instante de tiempo, el problema vuelve a ser resuelto, y se repite el proceso. Esta técnica se conoce como horizonte deslizante o, más comúnmente conocido, *receding horizon* [16].

Entre las características de MPC destacan las siguientes para la gestión de comunidades energéticas:

- Mecanismo de retroalimentación: incluye un mecanismo de retroalimentación que permite a los controladores gestionar de manera efectiva las imprecisiones y perturbaciones en tiempo real, lo que es crucial en entornos donde las perturbaciones son elementos inherentes.
- Multiobjetivo: otorga la posibilidad de no sólo resolver un problema de seguimiento, sino que también puede establecer parámetros económicos en la función a minimizar, sin obviar los esfuerzos de control.
- Gestión de restricciones: proporciona una gestión eficaz de las restricciones en tiempo real, lo que garantiza el cumplimiento de éstas dentro de límites especificados y cumpla con los requisitos operativos.

Dentro del dominio de las microrredes, los enfoques de MPC que tienen en cuenta las perturbaciones, en particular los métodos estocásticos [45],[41], han sido objeto de un interés significativo. Estos enfoques se dividen principalmente en dos grupos [13]:

- Métodos Analíticos: los métodos analíticos utilizan conceptos estadísticos para reformular tanto las restricciones como la función objetivo. Por ejemplo, un MPC estocástico para la gestión de una microrred podría modelar las previsiones de carga y energía eólica como distribuciones normales [32].
- Métodos Basados en Escenarios: en contraste, los métodos basados en escenarios asumen que las incertidumbres que afectan al sistema pueden describirse mediante un conjunto discreto de realizaciones futuras de dichas perturbaciones. Estas realizaciones son utilizadas posteriormente para calcular la secuencia óptima de acciones de control en función de los escenarios considerados. Este enfoque, debido a su naturaleza intuitiva, ha encontrado una amplia aplicación en la gestión de microrredes, como por ejemplo en [78], donde se propone un marco de MPC basado en escenarios para la programación y el control de una microrred de cogeneración con fuentes de energía renovable a gran escala.

4.2.1 Formulación MPC estándar

La formulación de un controlador MPC genérico puede ser definida de la siguiente forma:

$$\begin{aligned}
 \underset{\mathbf{U}}{\text{minimizar}} \quad & J = \sum_{k=0}^{N-1} J_k(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)) \\
 \text{sujeto a} \quad & \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)), \forall k \in \{0, 1, \dots, N-1\}, \\
 & \mathbf{x}(0) = \mathbf{x}_0, \\
 & \mathbf{x}(k) \in \mathcal{X}, \mathbf{u}(k) \in \mathcal{U}, \mathbf{w}_k \in \mathcal{W}, \\
 & \mathbf{u}(k) \in \mathcal{U}_k, \forall k \in \{0, 1, \dots, N-1\},
 \end{aligned} \tag{4.1}$$

donde J es la función objetivo a minimizar durante el horizonte de control, N es el horizonte de predicción y P_d denota la probabilidad de cada escenario; x es el estado del sistema, u es el vector de control, y w son las perturbaciones; la función \mathbf{f} representa el modelo del sistema; x_0 es el estado inicial; y los conjuntos \mathcal{X} y \mathcal{U}_k son las restricciones a las que están sujetos los estados y variables de control, respectivamente.

4.2.2 Formulación MPC basado en escenarios

Centrando la atención en los métodos basados en escenario, el método más sencillo y directo es el MPC-multiescenario (MS-MPC). En este enfoque, se aborda un único problema de control, pero se tiene en

cuenta una serie de escenarios que representan diferentes posibilidades de evolución del sistema debido a la incertidumbre. Estos escenarios se utilizan para calcular una única secuencia de control óptimo. Este enfoque, aunque efectivo, puede resultar en un rendimiento conservador debido a que escenarios muy improbables pueden influenciar mucho el comportamiento del controlador, dependiendo de las restricciones definidas. Un controlador MS-MPC puede ser definido de la siguiente forma:

$$\begin{aligned}
\underset{U}{\text{minimizar}} \quad & J = \sum_{k=0}^{N-1} \sum_{d=1}^{N_s} P_d [J_k(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}_d(k))] \\
\text{sujeto a} \quad & x(k+1) = f(x(k), u(k), w_d(k)), \forall k \in \{0, 1, \dots, N-1\}, \\
& x(0) = x_0, \\
& x(k) \in \mathcal{X}, u(k) \in \mathcal{U}, w_d(k) \in \mathcal{W}, \\
& u(k) \in \mathcal{U}_k, \forall k \in \{0, 1, \dots, N-1\},
\end{aligned} \tag{4.2}$$

donde N_s es el número de escenarios de perturbación, y P_d denota la probabilidad de cada escenario.

4.2.3 MPC basado en árboles

Una alternativa para otorgar al controlador de mayor flexibilidad es la variante de MPC basado en árbol de escenarios (más conocido como Tree Based MPC, TB-MPC). En TBMPC, los posibles escenarios se estructuran en un árbol, donde las bifurcaciones representan las diferentes realizaciones de la incertidumbre en el horizonte de predicción. La inclusión de restricciones no anticipativas [43] asegura que las secuencias de control en el árbol sean indistinguibles hasta que se alcancen las bifurcaciones propias del árbol de perturbaciones en el horizonte de predicción, tal y como se puede observar en las Figura 4.1 y 4.2, donde se aprecian los árboles de perturbación y de control, respectivamente. Este enfoque cuenta con la ventaja de que la influencia aquellos escenarios más improbables y que más pueden condicionar al controlador, quedan limitados a los instantes iniciales donde aún no se han producido bifurcaciones. De modo que escenarios remotos tendrán una menor influencia en la secuencia de control óptima que se acabe implementando. Sin embargo también ha de tenerse en cuenta que en TB-MPC se calcula una secuencia de control óptima para cada escenario considerado, de modo que hay que tener en cuenta el incremento de carga computacional.

Una alternativa que brinda mayor flexibilidad al controlador es el Control Predictivo basado en Árbol de Escenarios, conocido como Tree-Based MPC (TBMPC). En este enfoque, los posibles escenarios se organizan en una estructura de árbol, donde las bifurcaciones representan diferentes realizaciones de la incertidumbre en el horizonte de predicción. Para garantizar la coherencia entre las secuencias de control en el árbol, se aplican restricciones no anticipativas [43], tal y como se puede ver en las Figuras 4.1 y 4.2, donde se ilustran los árboles de perturbación y de control, respectivamente. La principal ventaja de esta aproximación radica en su capacidad para minimizar el impacto de escenarios improbables y distantes. Estos escenarios tienen un efecto limitado en la secuencia de control óptima que finalmente se implementa, ya que su influencia se restringe a los primeros instantes, antes de que ocurran las bifurcaciones. Por estos motivos, el controlador TBMPC demuestra ser una herramienta muy interesante en aplicaciones donde la incertidumbre desempeña un papel significativo, como la gestión de microrredes y comunidades energéticas. Su capacidad para lidiar con escenarios improbables y su adaptabilidad para abordar las complejidades de los recursos energéticos distribuidos, como paneles solares, vehículos eléctricos y baterías, lo hacen particularmente idóneo para estas aplicaciones. La formulación completa del controlador TB-MPC junto a sus restricciones no anticipativas será presentada en secciones posteriores.

4.2.4 MPC distribuido en comunidades energéticas

En el contexto de comunidades energéticas, donde múltiples agentes están separados geográficamente y poseen una variedad de recursos energéticos distribuidos (como paneles solares, vehículos eléctricos y baterías), la necesidad de coordinar y optimizar la gestión de energía se vuelve fundamental. La arquitectura de control centralizado, que puede ser eficaz en sistemas energéticos convencionales, no puede ser aplicada en estas comunidades donde cada hogar debe tener capacidad de decisión. Por ello, se requieren enfoques distribuidos que permitan a cada agente tomar sus propias decisiones de control y al mismo tiempo garantizar la coordinación y optimización global del sistema.

El control predictivo basado en modelo ha demostrado ser una técnica eficaz para abordar los desafíos asociados con la gestión de recursos energéticos distribuidos en comunidades energéticas. Sin embargo, para que MPC sea aplicable en este contexto, es necesario adaptar su arquitectura para satisfacer las necesidades

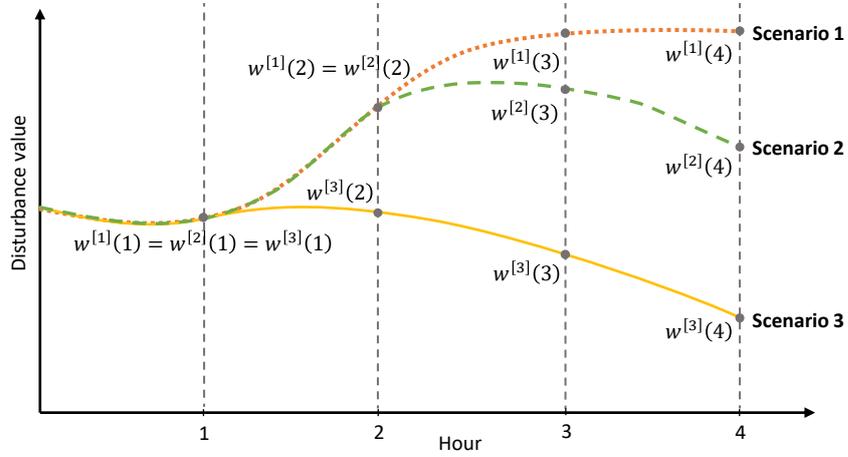


Figura 4.1 Árbol de perturbaciones.

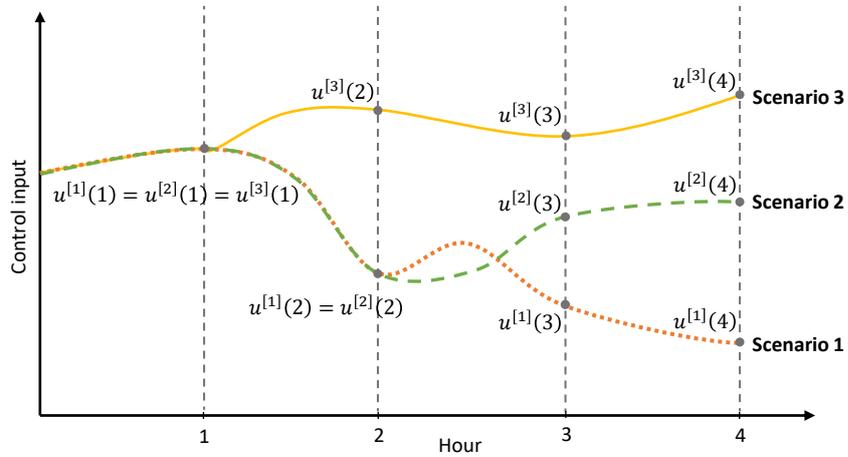


Figura 4.2 Árbol de control.

de un entorno distribuido. Para conseguir esto, se introduce el algoritmo de *Feasible cooperation based MPC* [75] (FC-MPC). El concepto central de este enfoque consiste en establecer una función de coste global Φ , que evalúa el rendimiento del sistema en su totalidad, como una combinación convexa de las funciones de coste locales Φ_i de cada agente $i \in \mathcal{A}$. La función de coste global se define de la siguiente manera:

$$\Phi = \sum_{i=1}^n \alpha_i \Phi_i, \quad (4.3)$$

donde $\alpha_i > 0$ y $\sum_{i=1}^n \alpha_i = 1$. El algoritmo FC-MPC asegura la factibilidad global del sistema en iteraciones intermedias, lo que significa que incluso si el tiempo de cómputo excede el tiempo requerido para la convergencia, la secuencia de control calculada en la última iteración sigue siendo factible para todo el

Algorithm 1: Given $(\mathbf{u}_i^0, x_i(k)) \quad \mathbb{Q}_i \geq 0, \mathbb{R}_i \geq 0, i \in \{1, M\}$ $p_{\max}(k) \geq 0$ and $\epsilon > 0$
 $p \leftarrow 1, \rho_i \leftarrow \Gamma\epsilon, \Gamma \gg 1$
while $\rho_i > \epsilon$ for some $i \in \{1, M\}$ and $p \leq p_{\max}$
do $\forall i \in \{1, M\}$
 $\mathbf{u}_i^{p(*)} \in \arg(\text{FC-MPC}_i)$, (see (9), (10))
end (do)
for each $i \in \{1, M\}$
 $\mathbf{x}_i^{p(*)} \leftarrow \mathbf{x}_i^{(\mathbf{u}_i^{p(*)}, \mathbf{u}_{j \neq i}^{p(*)}; x(k))}$
 $(\mathbf{x}_i^p, \mathbf{u}_i^p) = \frac{1}{M}(\mathbf{x}_i^{p(*)}, \mathbf{u}_i^{p(*)}) + (1 - \frac{1}{M})(\mathbf{x}_i^{p-1}, \mathbf{u}_i^{p-1})$
 $\rho_i = \|(\mathbf{x}_i^p, \mathbf{u}_i^p) - (\mathbf{x}_i^{p-1}, \mathbf{u}_i^{p-1})\|$
end (for)
 $p \leftarrow p + 1$
end (while)

Figura 4.3 Algoritmo genérico FC-MPC [75].

sistema. De forma genérica, el algoritmo FC-MPC consiste en que cada agente $i \in \mathcal{A}$ resuelva, durante un número de iteraciones p , el siguiente problema:

$$U_i^* \in \arg(\text{FC-MPC})_i, \quad (4.4)$$

donde FC-MPC_{*i*} se define como:

$$\text{FC-MPC}_i \triangleq \min_{u_i} \frac{1}{N} \Phi_i, \quad (4.5)$$

sujeto a las restricciones del modelo. Las secuencias de control óptimo se almacenan en U_i^* , y las variables de estado y entradas se denotan como x_i y u_i . Este modelo optimiza únicamente las variables asociadas a cada agente, manteniendo el resto de las variables constantes en comparación con la iteración anterior.

De manera muy resumida, el proceso de resolución del algoritmo FC-MPC se lleva a cabo en varias iteraciones, como se ilustra en la Figura 4.3, donde el objetivo es asegurar que cada agente calcule un par de secuencias de control y estado $(x_i^p; U_i^p)$ que difieran en una cantidad inferior a un error admisible μ con respecto a las calculadas en la iteración previa.

4.3 Papel de blockchain

El algoritmo distribuido FC-MPC requiere la comunicación entre los agentes y la ejecución de ciertas acciones de control. Para lograr esto, la tecnología blockchain en conjunción con los contratos inteligentes emerge como una solución que proporciona trazabilidad y la capacidad de auditar el proceso, sirviendo como la capa superior de un esquema distribuido jerárquico. Esta estructura de dos capas se compone de los hogares individuales en la capa inferior, que resuelven el problema TBMPC_{*i*} y envían sus resultados a la capa superior, representada por un contrato inteligente que lleva a cabo las siguientes funciones:

- Manejo del árbol de perturbaciones.
- Facilitación del intercambio de información entre agentes.
- Señalización de si los agentes pueden avanzar a la siguiente fase del algoritmo.

Es importante destacar que el proceso de ejecución de los contratos inteligentes no implica ningún tipo de centralización: al llamar a una función del contrato inteligente, dependiendo del tipo de mecanismo de consenso, se inicia un proceso completamente distribuido en el que múltiples nodos verifican que la ejecución de la función es válida. En otras palabras, todos los nodos que participan en el proceso de consenso obtienen el mismo resultado de la ejecución de la función.

La implementación real del contrato inteligente dentro de la plataforma energética será presentada en la sección 5.

5 Plataforma de gestión energética diseñada

En este capítulo se describirá la comunidad energética considerada, el algoritmo de control distribuido utilizado, y la implementación blockchain.

5.1 Formulación de la comunidad energética

La comunidad energética está compuesta por $N > 1$ hogares y se define mediante un grafo dirigido $\mathcal{G} = (\mathcal{N}, \mathcal{V})$, donde $\mathcal{N} = \{1, \dots, N\}$ es el conjunto de nodos, y el conjunto $\mathcal{V} \subseteq \mathcal{N} \times \mathcal{N}$ contiene conexiones dirigidas (i, j) desde el nodo i al nodo j en \mathcal{N} . Los nodos se dividen en el subconjunto de prosumidores $\mathcal{P} \subseteq \mathcal{N}$ y el subconjunto de consumidores $\mathcal{C} \subseteq \mathcal{N}$, con $\mathcal{N} = \mathcal{P} \cup \mathcal{C}$ y $\mathcal{P} \cap \mathcal{C} = \emptyset$. Los primeros involucran hogares con acceso a instalaciones de generación de energía y/o almacenamiento, como baterías o paneles solares, mientras que los consumidores no presentan ningún recurso energético distribuido y desempeñan un papel pasivo en la red. Para facilitar la comprensión de la notación, se añade la Tabla 5.1.

La gestión de energía se considera durante un horizonte de programación denominado T , dividido en N_h intervalos de tiempo de duración Δt , e indexados por $t = \{0, \dots, N_h - 1\}$. Cada hogar debe satisfacer una demanda de energía específica $p_{i,t}^1$ utilizando energía fotovoltaica, energía de la batería y/o importando energía de la red externa $p_{i,t}^g$ a través del punto de acoplamiento común (PCC), que presenta la siguiente función de coste:

$$\zeta_{i,t}(p_{i,t}^g) = c_t p_{i,t}^g \quad \forall i, t, \quad (5.1)$$

donde c_t representa el costo monetario de comprar energía del sistema de distribución en el tiempo t . No se considera la opción de vender energía a la red en este estudio. Los componentes que se pueden encontrar en cada nodo y una descripción detallada de cada uno de ellos se proporcionan a continuación.

Paneles solares

La generación fotovoltaica (PV) $p_{i,t}^{pv}$ para el agente $i \in \mathcal{N}$ en el tiempo t se describe de la siguiente manera:

$$p^{pv}i,t = \Psi_{i,t} A_i \eta^{pv} \text{gen}, \quad \forall i, t, \quad (5.2)$$

$$0 \leq \Psi_{i,t} \leq \bar{\Psi}_i, \quad \forall i, t, \quad (5.3)$$

donde $\Psi_{i,t}$ es la irradiancia global horizontal en la ubicación del agente i , A_i es la suma de las áreas de los paneles solares instalados por el agente i ($A_i = 0$ si $i \in \mathcal{C}$), y $\eta^{pv} \text{gen}$ es la eficiencia de conversión de los paneles solares. Se asume que la irradiancia global horizontal es la misma para todos los hogares debido a su cercanía geográfica, es decir, $\Psi_{i,t} = \Psi_t$. La potencia generada se asume que fluye directamente a las baterías, pero en casos en los que la producción de PV excede la demanda, el Sistema de Gestión de Baterías (también conocido como Battery Management System, BMS) corta la conexión entre los paneles solares y la batería. Este escenario solo ocurre cuando las baterías están completamente cargadas, la demanda de carga está satisfecha y no hay opciones de comercio de energía disponibles. En tales casos, la variable $p_{i,t}^{pvs}$ representa la energía excedente de $p_{i,t}^{pv}$ que no se puede almacenar, utilizar ni comercializar, es decir, $p_{i,t}^{pvs}$

Tabla 5.1 Nomenclatura del modelo de la comunidad energética.

Nomenclature	
Variables	
p^g	Power imported from the utility grid [kW]
E^b	Energy stored in a battery [kWh]
p^{pvs}	PV surplus power [kW]
p^{bc}	Battery charging power [kW]
p^{bd}	Battery discharging power [kW]
p^{bt}	Battery trading power [kW]
p^{ev}	EV charging power [kW]
Parameters	
c_t	Electricity pricing [€/kWh]
p^l	Fixed load demand [kW]
p^{pv}	PV power generation [kW]
η_c^b	Battery charging efficiency
η_d^b	Battery discharging efficiency
η_t^b	Battery trading efficiency
δ^b	Battery availability
η_c^{pv}	PV converter efficiency
δ^{ev}	EV charging availability
η_c^{ev}	EV charging efficiency
E_{ref}^{ev}	EV daily charge [kWh]
F	Trade matrix
Ψ	Global horizontal irradiance [W/m^2]
$p^{[k]}$	Probability associated to scenario k
Indices and notations	
$\cdot^{[k]}$	Value at discrete time k
$\tilde{\cdot}$	Aggregated arrays
$\bar{\cdot}, \underline{\cdot}$	Upper/lower boundary
$\cdot^{[k],p}$	Value of scenario k at iteration p
i, j	Agent indices
t	Hour index

funciona como una variable de holgura. Esta energía excedente se limita de la siguiente manera:

$$p_{i,t}^{pvs} \geq 0, \quad \forall i, t. \quad (5.4)$$

Baterías

La energía almacenada en una batería es el resultado de la carga y descarga de la batería por parte del agente $i \in \mathcal{N}$, el intercambio de energía con el resto de los agentes y la contribución del grupo fotovoltaico. Se representa como $E_{i,t}^b$ y se calcula de la siguiente manera:

$$E_{i,t}^b = E_{i,t-1}^b + (\eta_c^b p_{i,t}^{bc} - \frac{p_{i,t}^{bd}}{\eta_d^b} - \sum_{j \neq i}^N \frac{p_{ij,t}^{bt}}{\eta_t^b} + \eta_c^b (p_{i,t}^{pv} - p_{i,t}^{pvs})) \Delta t, \quad \forall i, t, \quad (5.5)$$

donde η_c^b , η_d^b y η_t^b representan las eficiencias de carga, descarga e intercambio de energía de la batería, respectivamente; $p_{i,t}^{bc}$, $p_{i,t}^{bd}$ y $p_{ij,t}^{bt}$ representan las potencias de carga, descarga e intercambio de baterías, respectivamente. Es importante destacar que en esta formulación, η_t^b se define para tener en cuenta las pérdidas de transmisión de energía durante el comercio, que se asumen linealmente dependientes de la potencia intercambiada, es decir, $\eta_t^b < \eta_d^b$. $p_{i,t}^{bc}$ se refiere a la potencia que se alimenta a la batería y no proviene de paneles solares, $p_{i,t}^{bd}$ es la potencia utilizada por el agente i para satisfacer su demanda, y $p_{ij,t}^{bt}$ representa la potencia enviada por el agente i a j en el tiempo t . Se aplican las siguientes restricciones a $E_{i,t}^b$, $p_{i,t}^{bc}$, $p_{i,t}^{bd}$ y $p_{ij,t}^{bt}$:

$$\underline{E}_{i,t}^b \delta_i^b \leq E_{i,t}^b \leq \bar{E}_{i,t}^b \delta_i^b, \quad \forall i,t, \quad (5.6)$$

$$0 \leq p_{i,t}^{bc} \leq \bar{p}_i^{bc} \delta_i^b, \quad \forall i,t, \quad (5.7)$$

$$0 \leq p_{i,t}^{bd} \leq \bar{p}_i^{bd} \delta_i^b, \quad \forall i,t, \quad (5.8)$$

$$0 \leq p_{ij,t}^{bt} \leq \bar{p}_{ij}^{bt} \delta_i^b, \quad \forall i,j,t, \quad (5.9)$$

donde el parámetro binario δ_i^b solo toma el valor 1 si el agente $i \in \mathcal{P}$.

Vehículos eléctricos

Los vehículos eléctricos (VE) se modelan como cargas controlables, y el conjunto $\mathcal{E} \subseteq \mathcal{N}$ contiene agentes que poseen un VE. El estado de carga del VE $E_{i,t}^{ev}$ está restringido y se calcula de la siguiente manera:

$$E_{i,t}^{ev} = E_{i,t-1}^{ev} + \eta_c^{ev} p_{i,t}^{ev} \Delta t, \quad \forall i,t, E_{i,t}^{ev} \leq E_{i,t}^{ev} \leq \bar{E}_{i,t}^{ev}, \quad \forall i,t, \quad (5.10)$$

donde $p_{i,t}^{ev}$ y η_c^{ev} representan la potencia de carga y la eficiencia de carga, respectivamente. Se define un parámetro binario $\delta_{i,t}^{ev}$ para expresar los instantes en los que se permite cargar los VE:

$$0 \leq p_{i,t}^{ev} \leq \delta_{i,t}^{ev} \bar{p}_i^{ev}, \quad \forall i,t, \quad (5.11)$$

donde \bar{p}_i^{ev} es el límite superior de carga de VE. Es importante destacar que si $i \notin \mathcal{E}$, entonces $\delta_{i,t}^{ev}$ siempre es igual a 0 $\forall t$. Por otro lado, si $i \in \mathcal{E}$, entonces $\delta_{i,t}^{ev}$ toma valores fijos que se determinan antes resolver el problema de optimización.

Se supone que cada VE debe recibir una carga diaria de energía de E_{ref}^{ev} . Esto se logra forzando que $E_{i,t}^{ev}$ sea igual a E_{ref}^{ev} a medianoche, es decir, a $T_\phi = 24$:

$$E_{i,T_\phi}^{ev} = E_{ref}^{ev}, \quad \forall i \in \mathcal{E}. \quad (5.12)$$

Además, se imponen restricciones adicionales a la carga diaria de energía de los VE después de la medianoche. Si κ es el número de horas en un día, es decir, $\kappa = 24$, se implementa la siguiente restricción terminal:

$$E_{i,\xi}^{ev} = (\theta + \frac{\xi}{\kappa}) E_{ref}^{ev} - \chi, \quad (5.13)$$

$$\chi = \max(0, \frac{\xi}{\kappa} E_{ref}^{ev} - \sum_{t=1}^{\xi} \bar{p}_i^{ev} \delta_{i,t}^{ev}), \quad (5.14)$$

donde ξ corresponde al último paso de tiempo contenido dentro del horizonte de planificación, y θ representa el número de días que transcurren desde el primer paso de tiempo del horizonte de planificación hasta el último. Por ejemplo, si la simulación comienza a las 5:00 a. m. el día 190, la longitud del horizonte de planificación es 24 y el tiempo de muestreo es de 1 hora, el último instante contenido en el horizonte es el día 191, y, por lo tanto, θ toma el valor 1. La ecuación (5.13) determina la carga deseada del VE en el último instante del horizonte; sin embargo, debe tener en cuenta la posibilidad de que el VE no pueda cargarse lo suficiente en ese último día debido a restricciones físicas, es decir, que el horario de carga del VE no permita suficiente carga. Si algún VE requiere una carga mayor de la que es físicamente posible, la variable χ se encarga de prevenir este tipo de infeasibilidad calculando la relajación necesaria de (5.13) basada en el número de intervalos de tiempo $\delta_{i,t}^{ev}$ disponibles para cargar los VE y la potencia máxima de carga por hora de los VE \bar{p}_i^{ev} .

Balances de potencia

El balance de potencia para cada agente determina la potencia comprada a la red eléctrica:

$$p_{i,t}^g = p_{i,t}^{bc} - p_{i,t}^{bd} - \sum_{j \neq i}^N p_{ji,t}^{bt} + p_{i,t}^l + p_{i,t}^{ev} \quad \forall i,t, \quad (5.15)$$

$$p_{i,t}^g \geq 0 \quad \forall i,t, \quad (5.16)$$

donde $0 \leq p_{i,t}^1 \leq \bar{p}_i^1$.

Es importante destacar que $\sum_{j \neq i}^N p_{j,t}^{b_i}$ representa la potencia recibida por el agente i del resto de los agentes.

Representación en espacio de estados

Considerando todas las ecuaciones mencionadas, podemos definir el siguiente modelo de espacio de estados lineal para la comunidad energética:

$$\begin{bmatrix} \tilde{E}_{t+1}^b \\ \tilde{p}_{t+1}^g \\ \tilde{E}_{t+1}^{ev} \end{bmatrix} = \begin{bmatrix} I_N & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_N \end{bmatrix} \begin{bmatrix} \tilde{E}_t^b \\ \tilde{p}_t^g \\ \tilde{E}_t^{ev} \end{bmatrix} + \begin{bmatrix} \eta_c^b \Delta t I_N & -\frac{\Delta t}{\eta_d^b} I_N & -\frac{\Delta t}{\eta_b^b} F & -\Delta t I_N & \mathbf{0} \\ I_N & -I_N & -F & \mathbf{0} & I_N \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \eta_c^{ev} \Delta t I_N \end{bmatrix} \begin{bmatrix} \tilde{p}_t^{bc} \\ \tilde{p}_t^{bd} \\ \tilde{p}_t^{bt} \\ \tilde{p}_t^{pvs} \\ \tilde{p}_t^{ev} \end{bmatrix} + \begin{bmatrix} \eta_c^b \Delta t I_N & \mathbf{0} \\ \mathbf{0} & I_N \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{p}_t^{pv} \\ \tilde{p}_t^1 \end{bmatrix}, \quad (5.17)$$

donde \tilde{E}_t^b , \tilde{p}_t^g y $\tilde{E}_t^{ev} \in \mathbb{R}^{N \times 1}$ agregan $E_{i,t}^b$, $p_{i,t}^g$ y $E_{i,t}^{ev} \forall i$, respectivamente; I_N representa una matriz de identidad de N dimensiones; $F \in \mathbb{R}^{N \times N(N-1)}$ denota la matriz de comercio donde cada término es nulo excepto por $N-1$ elementos que toman el valor 1 en cada fila para habilitar el comercio de energía entre todos los hogares; \tilde{p}_t^{bc} , \tilde{p}_t^{bd} , \tilde{p}_t^{pvs} , y $\tilde{p}_t^{ev} \in \mathbb{R}^{N \times 1}$ agregan $p_{i,t}^{bc}$, $p_{i,t}^{bd}$, $p_{i,t}^{pvs}$ y $p_{i,t}^{ev} \forall i$, respectivamente, y $\tilde{p}_t^{bt} \in \mathbb{R}^{N \times N(N-1)}$ $p_{i,j,t}^{b_i} \forall i$; \tilde{p}_t^{pv} y $\tilde{p}_t^1 \in \mathbb{R}^{N \times 1}$ agregan $p_{i,t}^{pv}$ y $p_{i,t}^1 \forall i$, respectivamente.

Observación: la Ecuación (5.17) presenta la formulación dinámica del equilibrio de potencia, donde $p_{i,t}^g$ se incluye como una variable de estado. Esta representación se utiliza por conveniencia notacional, y debe notarse que formalmente, $p_{i,t}^g$ se consideraría una variable de salida en lugar de una variable de estado.

Para simplificar, el modelo de espacio de estados (5.17) se representará como:

$$x(t+1) = Ax(t) + Bu(t) + Dw(t), \quad (5.18)$$

donde los vectores apilados $x(t) = [\tilde{E}_t^b, \tilde{p}_t^g, \tilde{E}_t^{ev}]^T$, $u(t) = [\tilde{p}_t^{bc}, \tilde{p}_t^{bd}, \tilde{p}_t^{bt}, \tilde{p}_t^{pvs}, \tilde{p}_t^{ev}]^T$, y $w(t) = [\tilde{p}_t^{pv}, \tilde{p}_t^1]^T$ representan el estado del sistema, las acciones de control y las perturbaciones medibles, respectivamente.

5.2 Control predictivo particularizado en la comunidad energética

A continuación se presenta cómo sería la formulación de un controlador predictivo aplicado a la comunidad energética considerada. El objetivo del control es minimizar el costo económico de la comunidad energética. Para ello, la función objetivo se calcula de la siguiente manera:

$$J(U) = \sum_{t=T_0}^{T_{N_h-1}} \sum_{i=1}^N \zeta_{i,t}(p_{i,t}^g) \quad (5.19)$$

donde ζ se define en (5.1); N_h es el horizonte de predicción; y $U = [u(T_0), \dots, u(T_{N_h-1})]$. El controlador MPC calcula la secuencia de control óptima que minimiza $J(U)$ de la siguiente manera:

$$\begin{aligned}
 U^* &= \arg \min_U J(U) \\
 \text{s.t.} & \\
 (5.17) & \forall t \in \{T_0, \dots, T_{N_h-1}\}, \\
 (5.6), (5.10), (5.16) & \forall t \in \{T_1, \dots, T_{N_h}\}, \\
 (5.12) & t = T_\phi, \\
 (5.13) & t = T_{N_h}, \\
 (5.4), (5.7), (5.8), (5.9), (5.11) & \forall t \in \{T_0, \dots, T_{N_h-1}\},
 \end{aligned} \tag{5.20}$$

donde $U^* = [u(T_0), u(T_1), \dots, u(T_{N_h-1})] \in \mathbb{R}^{(4N+N(N-1)) \times N_h}$. es la secuencia de control óptima que se aplicará a lo largo del horizonte de predicción, $x_{T_0} \in \mathbb{R}^{3N \times 1}$ es el estado inicial del sistema y $w_t = [w_{T_0}, w_{T_2}, \dots, w_{T_{N_h-1}}] \in \mathbb{R}^{2N \times N_h}$ es una secuencia determinística de las perturbaciones esperadas del sistema. El problema (5.20) está sujeto a $5NN_h + 2N + 1$ restricciones de igualdad y $10NN_h + 2N(N-1)N_h$ restricciones de desigualdad. Como se mencionó anteriormente, solo se aplica $u(T_0)$. En el siguiente instante de tiempo, se actualiza el estado actual del sistema y las previsiones de las perturbaciones, y se resuelve nuevamente el problema planteado en (5.20).

5.3 Control predictivo basado en árboles particularizado en la comunidad energética

La comunidad energética cuenta con dos fuentes de incertidumbre: la generación de energía solar debido a que las previsiones meteorológicas no son deterministas; y los perfiles de consumo de los hogares, que también son variables estocásticas. Para abordar estas incertidumbres, se presenta un enfoque de control predictivo basado en árboles. Esta estrategia utiliza un árbol de perturbaciones que tiene en cuenta una serie de posibles trayectorias futuras de ambas fuentes de perturbación. Estas trayectorias son mejor predichas durante los primeros instantes de la previsión (el primer valor de perturbación es igual para todas las trayectorias en el primer paso del horizonte, ya que es medido), pero a medida que pasa el tiempo, pueden ocurrir divergencias que se materializan como puntos de bifurcación.

TBMPC utiliza un árbol de perturbaciones y resuelve (5.20) para cada secuencia de perturbaciones dentro de dicho árbol. Desde un punto de vista distribuido, cada escenario, referenciado por el superíndice $[k]$, podría ser resuelto por un grupo de agentes en paralelo [38]. En consecuencia, dado que hay N_s escenarios, el problema establecido en (5.20) debe resolverse N_s veces. Sin embargo, es necesario imponer restricciones adicionales, conocidas como restricciones no anticipativas [64], para garantizar la compatibilidad en el proceso de optimización. Por lo tanto, se define la variable binaria $\gamma^{[kh]}$ de la siguiente manera:

- $\gamma^{[kh]}(t)$ toma el valor 1 si las variables de control de los escenarios k y h deben ser iguales en el tiempo t .
- $\gamma^{[kh]}(t) = 0$ en caso contrario.

Para obtener un árbol de secuencias de control que replique el árbol de perturbaciones, $\gamma^{[kh]}(t)$ debe ser 1 en los pasos de tiempo en los que las trayectorias de perturbación k y h se superponen. Además, se pueden imponer relajaciones adicionales para reducir la carga computacional de TBMPC, como mantener constante la secuencia de control de cada escenario durante los últimos $N_{ct} < N_h$ instantes, es decir, utilizando un horizonte de control $N_c = N_h - N_{ct}$ para calcular la secuencia de control. En general, cuanto más largo sea el horizonte, menos efecto tendrá en la primera parte de la secuencia de control óptima (recordar que el resto de la secuencia se descarta). Para ello, se define la variable binaria ζ de la siguiente manera:

- $\zeta(t)$ toma el valor 1 para los últimos N_{ct} pasos de tiempo del horizonte T .
- $\zeta(t) = 0$ en caso contrario.

La versión basada en árboles del MPC presentada en la Sección ?? se puede formular de la siguiente manera:

$$\begin{aligned}
& \min_{U^{[1]}, \dots, U^{[N_s]}} \sum_{k=1}^{N_s} p^{[k]} J(U^{[k]}) \\
& \text{s.t.} \\
& (5.17)^{[k]}, \forall t \in \{T_0, \dots, T_{N_h-1}\}, \\
& x^{[k]}(T_0) = x_{T_0}, \\
& (5.6)^{[k]}, (5.10)^{[k]}, (5.16)^{[k]}, \forall t \in \{T_1, \dots, T_{N_h}\}, \\
& w^{[k]}(t) = w_t^{[k]}, \forall t \in \{T_0, \dots, T_{N_h-1}\}, \\
& (5.12)^{[k]}, t = T_\phi, \\
& (5.13)^{[k]}, t = T_{N_h}, \\
& (5.4)^{[k]}, (5.7)^{[k]}, (5.8)^{[k]}, (5.9)^{[k]}, (5.11)^{[k]}, \forall t \in \{T_0, \dots, T_{N_h-1}\}, \\
& \zeta(t)u^{[k]}(T_t) = \zeta(t-1)u^{[k]}(t-1), \forall t \in \{T_1, \dots, T_{N_h-1}\}, \\
& \gamma^{[kh]}(t)u(t)^{[k]} = \gamma^{[kh]}(t)u(t)^{[h]}, \forall t \in \{T_0, \dots, T_{N_h-1}\}, \\
& \forall k, h \in \{1, \dots, N_s\},
\end{aligned} \tag{5.21}$$

donde $w^{[k]} = [w_{T_1}^{[k]}, w_{T_2}^{[k]}, \dots, w_{T_{N_h}}^{[k]}]$ es una secuencia determinista de los valores actuales y esperados de perturbación para la k -ésima trayectoria, y $p^{[k]}$ es la probabilidad asignada a cada escenario. Nótese que el superíndice $[k]$ junto a una ecuación indica que todos los estados, variables de control y perturbaciones que aparecen en ella pertenecen al escenario k .

Esta estrategia TBMPC permite manejar de manera eficiente la incertidumbre en la generación de energía PV y en los perfiles de consumo de los hogares, y se aplica a través de un conjunto de escenarios. La capacidad de paralelizar la resolución de estos escenarios y la introducción de restricciones no anticipativas garantizan la compatibilidad en el proceso de optimización.

5.4 Control predictivo distribuido particularizado en la comunidad energética

El problema TBMPC (5.21) de la sección anterior necesita ser resuelto de manera distribuida. Para ello, se utiliza el algoritmo FC-MPC presentado en la Sección 4.2.4 [75]. En este enfoque, los agentes trabajan de manera iterativa para minimizar una función objetivo común, que es la suma ponderada de la función objetivo de cada agente. El problema de optimización TBMPC _{i} para cada agente $i \in \mathcal{N}$ se formula de la

siguiente manera:

$$U_i^* = \arg \min (\text{TBMPC}_i) \text{ donde} \quad (5.22)$$

$$\text{TBMPC}_i \triangleq \min_{U_i^{[1]}, \dots, U_i^{[N_s]}} \frac{1}{N} \sum_{k=1}^{N_s} p^{[k]} J(U_i^{[k]})$$

$$\text{s.t.} \quad (5.23)$$

$$(5.17)_i^{[k]}, \forall t \in \{T_0, \dots, T_{N_h-1}\},$$

$$x_i^{[k]}(T_0) = x_{T_0},$$

$$(5.6)_i^{[k]}, (5.10)_i^{[k]}, (5.16)_i^{[k]}, \forall t \in \{T_1, \dots, T_{N_h}\},$$

$$w_i^{[k]}(t) = w_i^{[k]}, \forall t \in \{T_0, \dots, T_{N_h-1}\},$$

$$(5.12)_i^{[k]}, t = T_\phi,$$

$$(5.13)_i^{[k]}, t = T_{N_h},$$

$$(5.4)_i^{[k]}, (5.7)_i^{[k]}, (5.8)_i^{[k]}, (5.9)_i^{[k]}, (5.11)_i^{[k]}, \forall t \in \{T_0, \dots, T_{N_h-1}\},$$

$$\zeta(t) u_i^{[k]}(t) = \zeta(t-1) u_i^{[k]}(t-1), \forall t \in \{T_1, \dots, T_{N_h-1}\},$$

$$\gamma^{[kh]}(t) u_i^{[k]}(t) = \gamma^{[kh]}(t) u_i^{[h]}(t), \forall t \in \{T_0, \dots, T_{N_h-1}\},$$

$$\forall k, h \in \{1, \dots, N_s\},$$

donde $U_i^* = [u_i^{[1]}, u_i^{[2]}, \dots, u_i^{[N_s]}] \in \mathbb{R}^{(3+N) \times N_h}$ agrega las secuencias de control óptimas de cada escenario para el agente i . Nótese que en $u_i^{[k]}(t)$, solo se optimizan y actualizan las variables de control que pertenecen a i , mientras que el resto, es decir, las variables que pertenecen a los agentes vecinos, son valores constantes que muestran los resultados de las variables de control de la última iteración.

El algoritmo TBDMPC se puede ver en el Algoritmo 1, donde el superíndice p se refiere a la p -ésima iteración, por ejemplo, $u_i^{[k],p}$ es la secuencia de control óptima dentro del escenario k para el agente i en la iteración p ; y la secuencia de estado x calculada a partir de la secuencia de entrada u y el estado inicial x_0 se denota como $x^{(u;x_0)}$. En resumen, el algoritmo hace que cada hogar optimice sus entradas considerando su efecto en el rendimiento general basado en la información proporcionada por sus vecinos en la iteración anterior hasta que se logra la convergencia.

Algorithm 1 TBDMPC algorithm

Given $u_i^{[k],0}, x_{T_0}, i \in \{1, N\}, k \in \{1, N_s\}, c_i \geq 0, p_{\max} \geq 0, \varepsilon > 0, p \leftarrow 1, \forall \rho_i^{[k],1} \gg 1$

while $\max(\rho_i^{[k],p}) > \varepsilon$ for some $i \in \mathcal{N}, k \in \{1, N_s\}$ and $p \leq p_{\max}$ **do**

for each household $i \in \mathcal{N}$ **do**

$U_i^* \in \arg \min(\text{TBMPC}_i)$, Eq. (5.22)

end for

for each household $i \in \mathcal{N}$ **do**

for each scenario $k \in \{1, N_s\}$ **do**

$x_i^{[k],p(*)} \leftarrow x_i^{[k]}(u_i^{[k]}, u_{j \neq i}^{[k]}; x_{T_0})$ in Eq. (5.20)

$x_i^{[k],p} = \frac{1}{N} x_i^{[k],p(*)} + (1 - \frac{1}{N}) x_i^{[k],p-1}$

$u_i^{[k],p} = \frac{1}{N} u_i^{[k]} + (1 - \frac{1}{N}) u_i^{[k],p-1}$

$\rho_i^{[k],p} = \max(\|x_i^{[k],p} - x_i^{[k],p-1}\|,$

$\|u_i^{[k],p} - u_i^{[k],p-1}\|)$

end for

end for

$p \leftarrow p + 1$

end while

The image shows a user interface with three main sections:

- Step 1. Evaluate Disturbance Set:** Contains a 'Current disturbance set' section with a 'Disturbance set evaluated?' label and two buttons: 'Check disturbance set' and 'Check evaluation status'.
- Step 2. Initialization Process:** Contains two sections. The first is 'Initial state submitted by all households?' with 'Energy community initial state:' and fields for 'ID:' and 'Initial state:', followed by buttons 'Submit initial state', 'Check initial state status', and 'Retrieve initial state'. The second is 'Initial control sequence submitted by all households?' with 'Energy community initial control sequence:' and fields for 'ID:' and 'Initial control sequence:', followed by buttons 'Submit initial control sequence' and 'Check initial control sequence status'.
- Step 3. Solve TBDMPC algorithm:** Contains an 'Iteration counter:' section with a 'Get iteration counter' button. Below it are labels for 'Optimal control sequence submitted by all households?', 'Energy community optimal control sequence:', 'Global demand:', 'Global surplus:', and 'Algorithm finished?: 0'. There are buttons 'Check control sequence status' and 'Retrieve control sequence'. Below these are input fields for 'Optimal control sequence:', 'ID:', and 'Error:', followed by buttons 'Submit optimal control sequence', 'Submit error', and 'Check algorithm status'.

Figura 5.1 Interfaz gráfica de usuario.

Observación: el problema (5.22) describe una arquitectura de TBDMPC que depende de un árbol de perturbaciones. Este árbol debe actualizarse y regenerarse en cada intervalo de tiempo después de analizar los valores actuales de perturbación. El nuevo árbol debe ser el mismo para todos los agentes, lo cual puede ser problemático en el caso de los subsistemas distribuidos. Este problema se puede abordar utilizando un controlador jerárquico. En este trabajo, la capa superior maneja la información de todos los agentes, calcula la raíz del árbol para el próximo intervalo de tiempo y la entrega a los subsistemas.

5.5 Desarrollo de la aplicación blockchain

El Algoritmo 1 requiere comunicación entre agentes, y se deben tomar ciertas acciones de control. Para lograr esto, el algoritmo se integra junto a un contrato inteligente desplegado en la red de Ethereum, lo que proporcionará trazabilidad y la capacidad de auditar el proceso, sirviendo como la capa superior de un esquema jerárquico distribuido. Esta estructura de dos capas está compuesta por hogares individuales en la capa inferior, que resuelven el problema $TBMPC_i$ y envían sus resultados a la capa superior.

En este estudio, Rinkeby [3], que es una red de prueba de Ethereum [15], es la red de blockchain elegida en la que se implementa y ejecuta el algoritmo TBDMPC. Ethereum es la blockchain seleccionada por ser el ecosistema más maduro que permite a los usuarios crear y desplegar contratos inteligentes gracias a Solidity [10], un lenguaje de programación Turing completo. El proceso de desarrollo de la aplicación se detalla a continuación, donde se proporciona una descripción completa de todos los recursos necesarios y se presenta un esquema que aclara las conexiones entre estos recursos en la Figura 5.2:

- Se desarrolla una interfaz gráfica (GUI) basada en *React* [9] para servir como punto de entrada para los hogares, de modo que los datos se puedan cargar y recuperar, como se puede ver en la Figura 5.1.
- Metamask es una billetera de criptomonedas y puerta de enlace a aplicaciones blockchain que proporciona una gestión segura de las cuentas y activos de ellas [6].
- Se requiere un nodo para interactuar con la blockchain. Esto lo proporciona Infura [5], que es un grupo de nodos Ethereum funcionales que permiten a los desarrolladores acceder a ellos sin tener que realizar una instalación completa.
- *Web3.js* [11] es una colección de bibliotecas que hacen posible la conexión a un nodo Ethereum local o remoto utilizando HTTP, IPC o WebSocket. En resumen, *Web3.js* permite la interacción con nodos individuales con llamadas de procedimiento remoto JSON (RPC) [14], [49].

Una vez el contrato inteligente es desplegado, se habilita la interacción con él y la ejecución de sus funciones. Sin embargo, dado que el algoritmo tiene varias fases, es necesario cuidar el diagrama de flujo del algoritmo

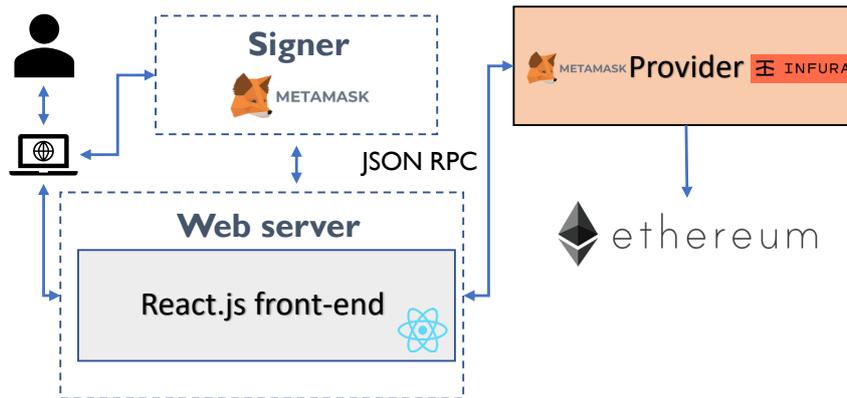


Figura 5.2 Arquitectura de la aplicación desarrollada.

y tomar ciertas decisiones. Para esto, se utiliza OpenZeppelin Defender [7]. Esta plataforma proporciona un entorno de operaciones de seguridad que pueden automatizar el diagrama de flujo de los contratos inteligentes sin complicar el código de programación del mismo. Los servicios de Zeppelin Defender utilizados en este trabajo son los siguientes:

- El servicio Defender Contract Sentinel permite el monitoreo de transacciones mediante el seguimiento de funciones específicas, eventos y parámetros de transacción.
- El servicio Defender Relay permite a los usuarios enviar transacciones mediante una API, encargándose de la firma de las transacciones y la estimación de las tarifas de la red, entre otras tareas [2].
- El servicio Defender Autotasks permite a los usuarios ejecutar automáticamente código en respuesta a transacciones.

La conjunción de sentinelas, relayers y autotasks permite una automatización completa del flujo de control del algoritmo mediante el monitoreo de eventos y la ejecución de autotareas por parte de los relayers. En la Figura 5.3, se muestra un esquema que ilustra la conexión entre los servicios de Defender y el contrato inteligente. A continuación, se presenta un ejemplo sencillo de este proceso:

1. El agente i envía U_i^* al contrato inteligente y espera a que el resto de los agentes envíen los suyos antes de pasar a la siguiente fase del algoritmo.
2. Supongamos que el agente j es el último en enviar su secuencia U_j^* al contrato inteligente. Después de esta presentación, se emite un evento desde el contrato inteligente a la blockchain.
3. Un centinela que monitorea eventos dentro de la blockchain detecta el evento y notifica a un relé sobre su presencia.
4. Al recibir la notificación, el relé activa una autotarea que ejecuta una función del contrato inteligente que modifica el estado de una variable pública.
5. Los agentes notifican a través de la GUI el valor actualizado de la variable pública, lo que significa que la fase actual del algoritmo ha concluido.

Con la inclusión de estas características, el Algoritmo 1 se modifica al Algoritmo 2, donde se destaca el papel del contrato inteligente, y de los hogares. Para simplificar la extensión del algoritmo, cada vez que se genera un evento, se asume que un centinela alertará inmediatamente a un relayer, que a su vez activará una tarea automática para gestionar eficazmente el flujo de control:

1. Antes de que comience la primera iteración, el contrato inteligente desempeña un papel fundamental: evalúa las perturbaciones actuales y determina qué conjunto de perturbaciones (DS, por sus siglas en inglés) coincide mejor con estos valores. Para clarificar el concepto de "conjunto de perturbaciones", es importante señalar que existen un total de $N_{CP} \times N_G$ DS (donde N_{CP} y N_G son el número de conjuntos de perturbaciones relativas a la carga y a la generación, respectivamente), y cada uno de ellos abarca una cantidad significativa de escenarios, específicamente $(N_{CP} \times N_G)^2$. Al ejecutar este paso, solo se selecciona y se trae a al problema de optimización uno de los $N_{CP} \times N_G$ DS.

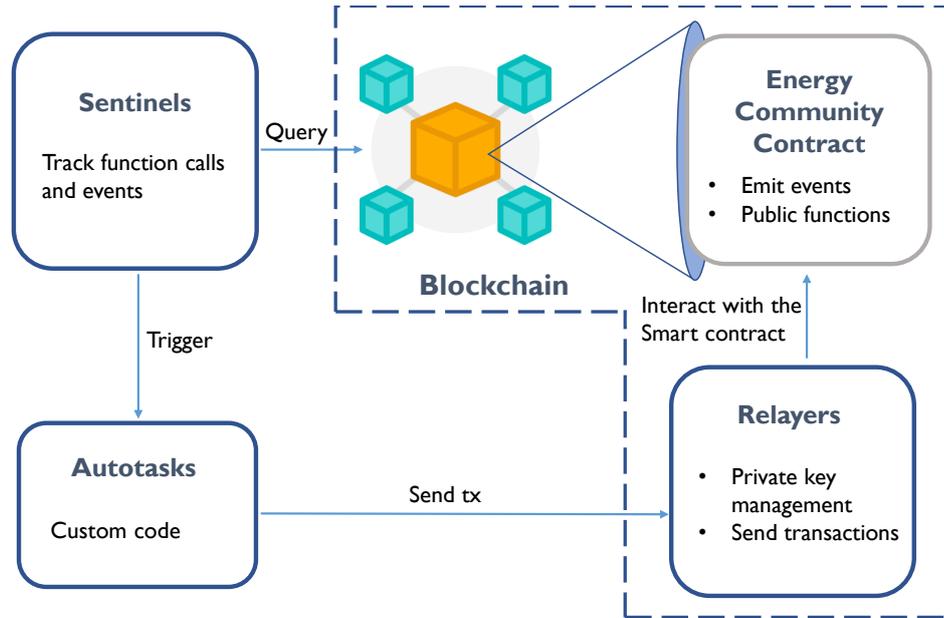


Figura 5.3 Integración de Defender Services junto al contrato inteligente.

2. La primera iteración comienza con cada agente, denotado como i , enviando su estado inicial, representado como $x_{i,T_0} = [E_{i,T_0}^b, p_{i,T_0}^g, E_{i,T_0}^{ev}]^T$, al contrato inteligente.
3. Una vez que todos los agentes en el conjunto \mathcal{N} han completado con éxito el paso 2, se genera y registra un evento.
4. Para generar una solución inicial factible, los agentes acceden a los estados de todos los demás agentes, $x_{j,T_0} \forall j \in \mathcal{N}$, y resuelven de manera independiente el problema centralizado de TB MPC (5.21). Es importante destacar que este problema se resuelve sin ningún intercambio de potencia, lo que significa que $\bar{p}_{ij}^{bl} = 0$ para todos los i y j en \mathcal{N} . La secuencia de control calculada, $U_i^{T_0}$, se envía posteriormente al contrato inteligente.
5. Después de cargar todas las secuencias de control iniciales de los N agentes en el contrato inteligente, se genera un evento.
6. Los agentes recuperan las secuencias de control $U_j^0 \forall j \neq i \in \mathcal{N}$ del contrato inteligente. Luego, aplican estas secuencias para resolver el problema $TBMPC_i$ y envían sus secuencias de control óptimas, denominadas U_i^* , de vuelta al contrato inteligente.
7. Tras completar con éxito el Paso 6, se genera y registra otro evento en la blockchain.
8. Cada agente i recupera las secuencias de control óptimas, $U_j^* \forall j \neq i \in \mathcal{N}$, del contrato inteligente. Posteriormente, utilizan esta información para calcular varias variables clave, incluyendo $x_i^{[k],p*}$, $x_i^{[k],p}$, u_i^p , y $\rho_i^{[k],p}$ para todos los k en el conjunto $\{1, N_s\}$.
9. Cada iteración p llega a su fin una vez que todos los agentes $i \in \mathcal{N}$ hayan enviado sus respectivos errores, $\rho_i^{[k],p}$ al contrato inteligente. Posteriormente, el contrato inteligente evalúa la condición de terminación, que especifica que

$$\max \rho_i^{[k],p} \leq \varepsilon \forall i \in \mathcal{N}, \forall k \in \{1, N_s\},$$

Si esta condición se cumple, finaliza el algoritmo; en caso contrario, comienza una nueva iteración. En ambos escenarios, se genera y almacena un evento en la cadena de bloques, indicando el estado del algoritmo o el inicio de una nueva iteración.

Algorithm 2 Blockchain-based TBDMPC algorithm

Given $c_t \geq 0, p_{\max} \geq 0, \varepsilon > 0, p \leftarrow 1, \forall \rho_i^{[k],1} \gg 1, i \in \mathcal{N}, k \in \{1, N_s\}$

while $\max(\rho_i^{[k],p}) > \varepsilon$ and $p \leq p_{\max}$ **do**

if $p = 1$ **then**

for each household $i \in \mathcal{N}$ **do**

 • Upload p_i^1 and p_i^{pv} to the smart contract.

end for

 The smart contract executes:

 • Evaluate $\forall p_i^1, p_i^{\text{pv}}$ for each household $i \in \mathcal{N}$ and determine the DS.

for each household $i \in \mathcal{N}$ **do**

 • Retrieve the DS.

 • Solve TBMPC_{*i*} setting $\bar{p}_{ij}^{\text{b}} = 0$.

 • Upload U_i^* and x_{i,T_0} to the smart contract.

 • Retrieve U_j^* and $x_{j,T_0} \forall j \neq i$ from the smart contract.

end for

 The smart contract executes:

for each household $i \in \mathcal{N}$ **do**

 • Collect U_i^* and x_{i,T_0} .

end for

else

for each household $i \in \mathcal{N}$ **do**

 • Solve TBMPC_{*i*}.

 • Upload U_i^* to the smart contract.

 • Retrieve $U_j^* \forall j \neq i$ from the smart contract.

for each scenario $k \in \{1, N_s\}$ **do**

 • Compute locally:

$$x_i^{[k],p(*)} \leftarrow x_i^{[k]}(u_i^{[k]}, u_{j \neq i}^{[k]}; x_{T_0}) \text{ in Eq. (5.20)}$$

$$x_i^{[k],p} = \frac{1}{N} x_i^{[k],p(*)} + \left(1 - \frac{1}{N}\right) x_i^{[k],p-1}$$

$$u_i^{[k],p} = \frac{1}{N} u_i^{[k]} + \left(1 - \frac{1}{N}\right) u_i^{[k],p-1}$$

$$\rho_i^{[k],p} = \max(\|x_i^{[k],p} - x_i^{[k],p-1}\|, \|u_i^{[k],p} - u_i^{[k],p-1}\|)$$

end for

 • Upload $\forall \rho_i^{[k],p}$ to the smart contract.

end for

 The smart contract executes:

for each household $i \in \{1, N\}$ **do**

 • Collect ρ_i^p .

end for

 • Verify the terminating condition.

 • $p = p + 1$.

end if

end while

6 Resultados y Simulaciones

Esta sección presenta un estudio de caso realista y evalúa los resultados del problema de optimización en bucle cerrado descrito en el Capítulo 5. En primer lugar, se proporciona una descripción del sistema simulado. En segundo lugar, se presenta la información necesaria sobre la construcción del árbol de perturbaciones y los parámetros de control. Finalmente, se ilustran y discuten los resultados de las simulaciones.

6.1 Caso de estudio

La topología de la comunidad de energía simulada se ilustra en la Figura 6.1. Como se puede ver, hay 10 hogares en la red (4 prosumidores con paneles solares y 6 consumidores). La ubicación geográfica de la microred residencial es Sevilla, España. El precio de la energía suministrada por la red externa se obtiene del mercado eléctrico ibérico (MIBEL). El coste de la electricidad importada de la red de servicios públicos se ilustra en la Figura 6.2. Detalles numéricos de los dispositivos que forman parte del modelo se muestran a continuación:

6.1.1 Batería

Se asume que todos los prosumidores tienen acceso a baterías. La capacidad de almacenamiento de cada batería de los prosumidores toma un valor aleatorio en el rango de [3.4,4] kWh; la cota superior e inferior del estado de carga (SOC) se establecen en 85 % y 30 %, respectivamente. Tanto las eficiencias de carga como de descarga se establecen en 94 %, y la eficiencia de comercio se fija en 92.5 %.

6.1.2 Paneles Solares Fotovoltaicos

Los prosumidores poseen un número aleatorio de paneles solares fotovoltaicos en el rango de 15 a 25, con una superficie nominal estándar de 1 m², y un rendimiento de conversión $\eta_{\text{gen}}^{\text{PV}}$ del 20 %.

6.1.3 Vehículo Eléctrico

La disponibilidad de carga de vehículos eléctricos T^{ev} se genera para cada propietario tal que $T_i^{\text{ev}} = [\delta_{i,T_1}^{\text{ev}}, \delta_{i,T_2}^{\text{ev}}, \dots, \delta_{i,T_{N_h}}^{\text{ev}}]$, y pueden ser observadas en la Figura 6.3. La energía diaria que debe ser suministrada a cada vehículo eléctrico es un valor aleatorio en el rango de [7.49 – 8.13] kWh. El límite superior de la potencia de carga es de $\bar{p}_i^{\text{ev}} = 2.5$ kW y la eficiencia asociada $\eta^{\text{ev}} = 0.92$ %.

6.2 Árbol de perturbaciones y parámetros de control

El horizonte de predicción N_h se establece en 24 horas, el horizonte de control N_c se establece en 20 horas, es decir, $N_{\text{ct}} = 4$, y el tiempo de muestreo se establece en 1 hora. Como se mencionó anteriormente, hay dos fuentes de perturbaciones en (5.17): la generación de energía fotovoltaica y la demanda de carga no controlable. Para cada fuente de perturbación, se utiliza un conjunto de datos base para generar el conjunto de posibles trayectorias futuras aplicando diferentes factores de escala:

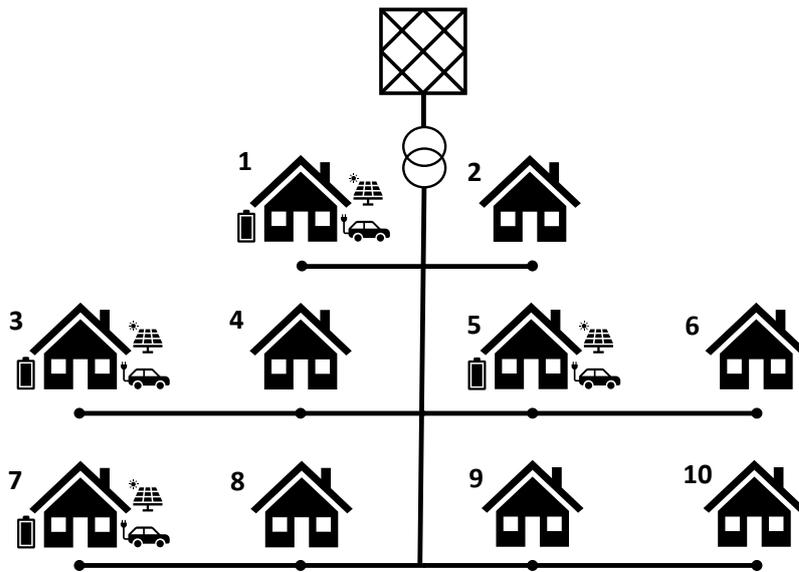


Figura 6.1 Comunidad de energía simulada.

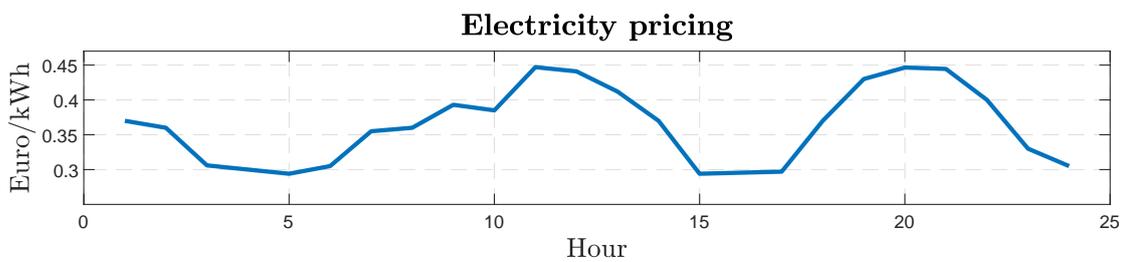


Figura 6.2 Precio al que se compra la electricidad de la red de servicios públicos.

- El conjunto de datos de perfiles de consumo base (BCP) para los hogares se genera a partir de un modelo estocástico [61]. El conjunto de datos contiene perfiles de demanda de múltiples hogares durante un año. Utilizando el BCP, se generan dos conjuntos de perfiles adicionales: un conjunto de perfiles de consumo bajo (LCP) y otro de perfiles de consumo alto (HCP).
- Para la generación de energía fotovoltaica, se utilizan datos del Sistema de Información Geográfica

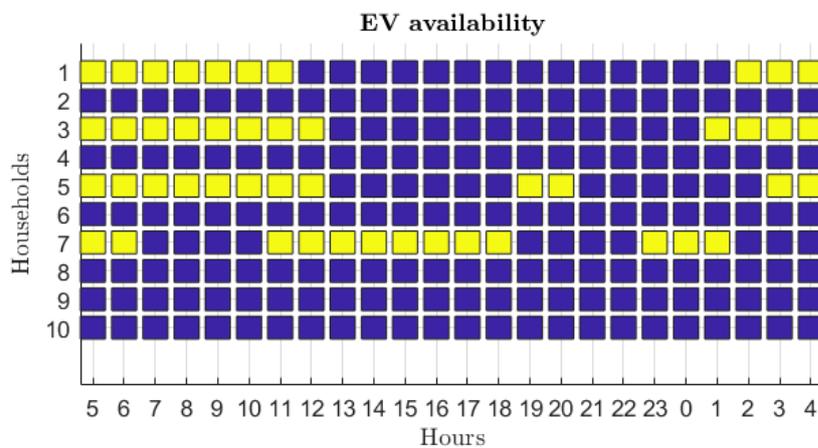


Figura 6.3 Disponibilidad de carga de vehículos eléctricos. Las casillas amarillas y azules representan los intervalos de tiempo de carga de vehículos eléctricos disponibles y no disponibles, respectivamente.

Fotovoltaica [8]. Específicamente, los datos de irradiancia global horizontal del año meteorológico típico (TMY). Los datos seleccionados se eligen de los datos horarios en un período de 10 años. Dado el conjunto de datos base de irradiancia horizontal global (BG), se generan otros dos conjuntos de datos adicionales: un conjunto de datos de irradiancia horizontal baja (LG) y otro conjunto de datos de irradiancia horizontal alta (HG).

Cada escenario se construye combinando pares de conjuntos de datos de la siguiente manera:

1. Dentro de cada escenario, se establecen dos puntos de bifurcación (BP) en los momentos $t = \{8, 14\}$ horas. Esto significa que los escenarios se dividen en tres secciones durante el horizonte de predicción, como se puede observar en la Figura 6.4. La selección de estos momentos de bifurcación se asume debido a la fuerte influencia que ambas fuentes de perturbaciones tienen en la hora del día, en particular durante las horas centrales.
2. Dado que existen $N_{CP} = 3$ conjuntos de datos de perfiles de consumo y $N_G = 3$ conjuntos de datos de irradiancia global, existen $N_{CP} \times N_G = 9$ combinaciones posibles dentro de cada sección. Dado que hay tres secciones en cada escenario, se pueden generar $(N_{CP} \times N_G)^3$ posibles escenarios de perturbación.
3. Cada sección del escenario k corresponde a la combinación de un solo conjunto de datos de perfil de consumo y un conjunto de datos de irradiancia horizontal global, por ejemplo, para $t = \{8, 9, \dots, 13\}$: $w^{[1]}(t) = [BG(t); LCP(t)]$, $w^{[2]}(t) = [MG(t); LCP(t)]$, $w^{[3]}(t) = [HG(t); LCP(t)]$, $w^{[4]}(t) = [BG(t); BCP(t)]$, \dots , $w^{[9]}(t) = [HG(t); HCP(t)]$.

Al analizar estas reglas, queda claro que se pueden generar 9 posibles árboles de perturbación o DS en cada instante, ya que la primera sección, es decir, la raíz del árbol, puede pertenecer a cualquiera de los 9 escenarios posibles, como se muestra en la Figura 6.5. Sin embargo, al medir los valores de las perturbaciones en el paso de tiempo actual, solo uno de los árboles es seleccionado por el contrato inteligente, mientras que el resto son descartados, es decir, el árbol de perturbación final está compuesto por $(N_{CP} \times N_G)^2 = 81$ escenarios. Para simplificar, la probabilidad asociada a cada escenario $p^{[k]}$ es $\frac{1}{N_{CP} \times N_G}$, y $\sum_{k=1}^{N_{CP} \times N_G} p^{[k]} = 1$.

Es importante destacar que, a pesar de comenzar con solo tres conjuntos de datos para irradiancia y perfiles de consumo, debido a que cada trayectoria futura contenida en el árbol está compuesta por tres secciones independientes y cada una de ellas puede pertenecer a cualquiera de los tres conjuntos de datos, se pueden agrupar un total de 3^3 realizaciones futuras diferentes de irradiancia y consumo.

6.3 Resultados de Simulación

6.3.1 Simulaciones

El día comienza a las 5 AM en cada simulación. Tanto la simulación en bucle cerrado como el horizonte de predicción N_h se configuran para representar un día con intervalos de tiempo de una hora ($\Delta t = 1$ h), lo que

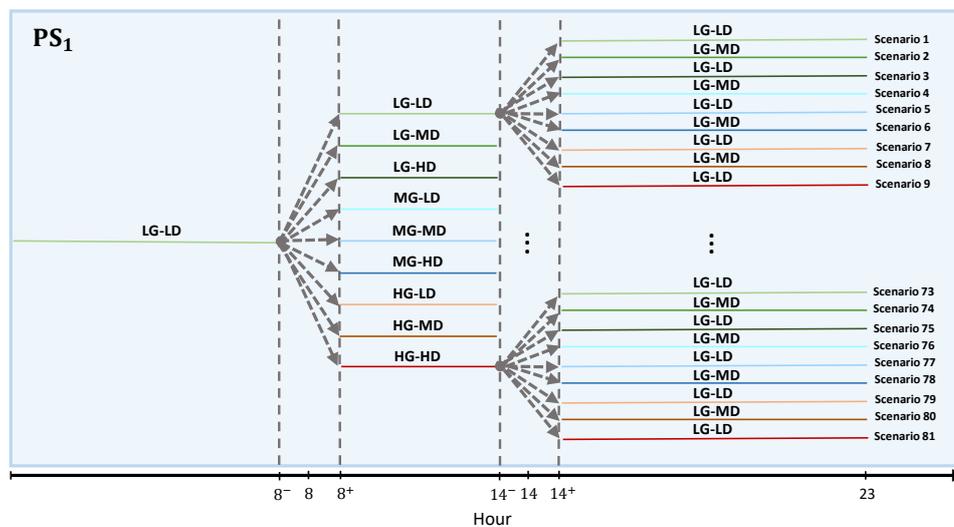


Figura 6.4 Evolución del árbol de perturbaciones a lo largo del horizonte de predicción.

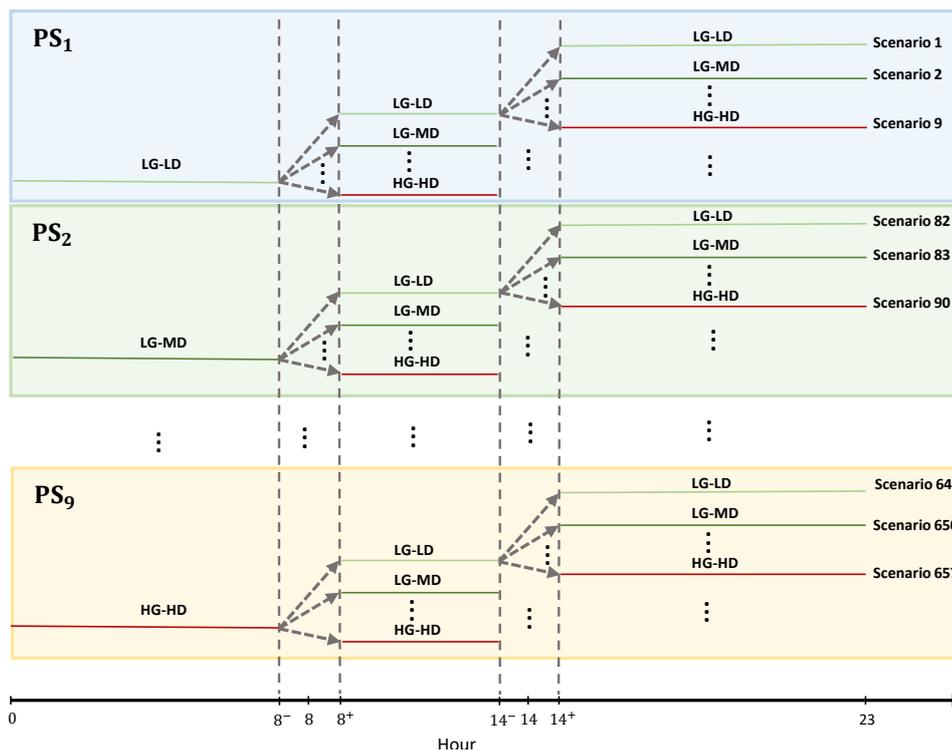


Figura 6.5 Conjunto de árboles de perturbación que pueden ser generados a las 0 AM.

lleva a un horizonte de programación inicial $T = [5, 6, 7, \dots, 24, 1, \dots, 4]$. El resto de esta sección muestra los resultados obtenidos mediante la implementación del algoritmo en las siguientes configuraciones de comercio de energía:

- Minimización del esfuerzo económico global de la comunidad (ME): la energía se comercia entre los prosumidores dentro de la red a una tasa de descuento del 100% con respecto a los precios de liquidación del mercado.
- Configuración de comercio estándar (BT): la energía se comercia a un descuento del 50% con respecto al mercado minorista de electricidad.
- Configuración sin comercio (NT): no se permite el comercio de energía.

Para cada una de las configuraciones mencionadas, se realizan dos simulaciones en diferentes estaciones. Los subíndices w y s se refieren a invierno y verano, respectivamente. La generación total de energía solar fotovoltaica y la demanda de carga fija de la comunidad para las simulaciones de verano e invierno se muestran en la Figura 6.6.

El rendimiento de cada simulación se evalúa según los índices de rendimiento económico y de uso de energía. La Tabla 1 muestra los costes monetarios tanto para los prosumidores como para los consumidores, y la Tabla 2 muestra los índices de rendimiento de uso de energía considerados en este estudio. Los índices de uso de energía mencionados son:

- Energía intercambiada entre hogares (K_1): el volumen de energía intercambiado entre hogares dentro de la comunidad.
- Energía importada de la red eléctrica (K_2): el volumen de energía comprado a la red eléctrica.
- Uso de baterías (K_3): el volumen de energía suministrado por las baterías.
- Uso de energías renovables (K_4): la relación entre el volumen de energía utilizada que se genera localmente y el volumen de energía que se genera localmente.

Los resultados de las simulaciones de la comunidad energética en verano se muestran en las Figuras (6.7, 6.9, 6.11) para los casos ME, BT y NT, respectivamente; y los resultados de las simulaciones en invierno se muestran en las Figuras (6.8, 6.10, 6.12). Los resultados de las simulaciones del comportamiento de las

baterías se muestran en las Figuras (6.13, 6.15, 6.17) para los casos ME, BT y NT, respectivamente; y los resultados de las baterías en invierno se muestran en las Figuras (6.14, 6.16, 6.18). Al comparar los resultados de las simulaciones de la comunidad energética ME, BT, y NT, en bucle cerrado, se puede observar una diferencia notable en términos de energía comerciada entre los diferentes escenarios en cada estación. Las curvas de comercio de energía ME_s y ME_w son significativamente más altas, a pesar de compartir una forma similar, que las producidas en los escenarios BT_s y BT_w , respectivamente. La compra de energía de la red eléctrica se reduce al mínimo en todos los escenarios ME y BT, especialmente durante las horas pico, ya que los prosumidores pueden mitigar no solo sus perfiles de consumo, sino también los de los consumidores gracias al uso de baterías y al exceso de generación de energía solar fotovoltaica. Siendo cargas flexibles, los vehículos eléctricos solo se cargan durante las franjas horarias más económicas o cuando hay una generación fotovoltaica elevada, de acuerdo con la disponibilidad de carga. Como se puede observar en las Figuras (6.13, 6.15, 6.17, 6.14, 6.16, 6.18), las baterías actúan como búferes de energía que se cargan cuando existe un exceso de generación de energía solar fotovoltaica o, de manera similar a los vehículos eléctricos, cuando la electricidad tiene un precio bajo. Se puede reconocer que la generación neta de energía solar fotovoltaica, que representa la diferencia entre la generación de energía solar fotovoltaica y el exceso de energía solar fotovoltaica, es significativamente menor en los escenarios NT_s y NT_w en comparación con los escenarios ME y BT, ya que no se permite el comercio de energía y los prosumidores no pueden gestionar el exceso de generación de energía solar fotovoltaica. Además, la potencia de carga de las baterías se vuelve más relevante en ausencia de generación de energía solar fotovoltaica, particularmente en los escenarios ME debido al mayor uso de baterías.

En cuanto a los costes económicos mostrados en la Tabla 6.1, se observa que los valores más altos corresponden a los escenarios NT_s y NT_w , con costes de 31.01 € y 30.13 €, respectivamente. Los costes económicos más bajos en cada temporada son obtenidos de los escenarios ME, que representan el 58.27% y el 86.49% de los costos del escenario NT en verano e invierno, respectivamente. Los escenarios BT incorporan una compensación económica por la energía comerciada dentro de la comunidad de energía, lo que resulta en diferencias notables en los costes para prosumidores y consumidores entre los escenarios ME y BT. Observando las simulaciones de verano y comparando los dos escenarios habilitados para el comercio, ME_s arroja costos promedio para los prosumidores y consumidores de 3.19 € y 0.88 €, respectivamente, mientras que el escenario BT_s muestra un coste económico de 2.42 € y 2.51 € para prosumidores y consumidores, respectivamente. También es digno de mención que los prosumidores presentan una demanda notablemente mayor que los consumidores debido a sus vehículos eléctricos. En ME, los prosumidores utilizan paneles solares y baterías para mitigar los costos económicos de toda la comunidad, lo que conlleva a costes elevados para los prosumidores (12.77 € y 18.94 € en verano e invierno, respectivamente). Esta carga se reduce en los

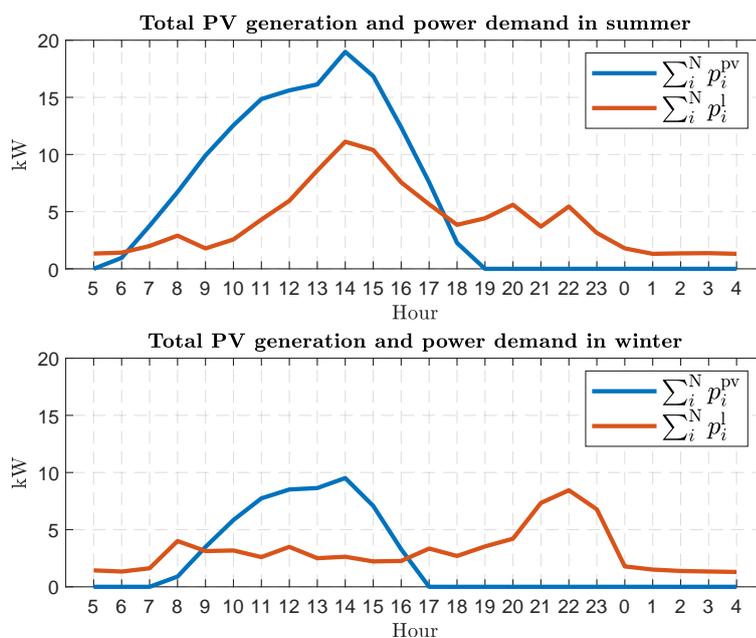


Figura 6.6 Generación total de energía solar fotovoltaica y demanda de carga fija en la comunidad de energía en verano e invierno.

escenarios BT, ya que los prosumidores son recompensados por la energía comerciada, con una reducción de sus costes del 24.35% y el 18.95% en verano e invierno, respectivamente, mientras que los costes de los consumidores aumentan un 184.53% y un 72.47% en verano e invierno, respectivamente. Los costes más bajos para los prosumidores son obtenidos lógicamente de los escenarios NT, ya que se centran únicamente en la reducción de sus costes individuales.

Al observar las métricas de uso de energía en la Tabla 6.2, se observa que los valores más altos de K_1 para cada temporada se encuentran en los escenarios ME, con valores 91.82% y un 176.84% más altos que en los escenarios BT_s y BT_w , respectivamente. En cambio, en los escenarios BT se observa un valor mucho mayor de K_2 , lo que significa que los prosumidores tienden a priorizar el autoconsumo en lugar de comerciar energía. También es muy reconocible el efecto de la elevada generación de energía solar fotovoltaica en verano con respecto a la obtenida en invierno: a medida que disminuye la disponibilidad de generación de energía solar fotovoltaica, también lo hace K_1 , e, inversamente, K_2 aumenta. En cuanto a K_4 , los escenarios habilitados para el comercio presentan mejores resultados debido a su capacidad para gestionar el exceso de energía solar fotovoltaica enviándola a sus vecinos. Sin embargo, es importante destacar que ningún escenario logra alcanzar un valor del 100% para K_4 , y valores más altos podrían lograrse con mayores demandas de energía o capacidades de almacenamiento más grandes.

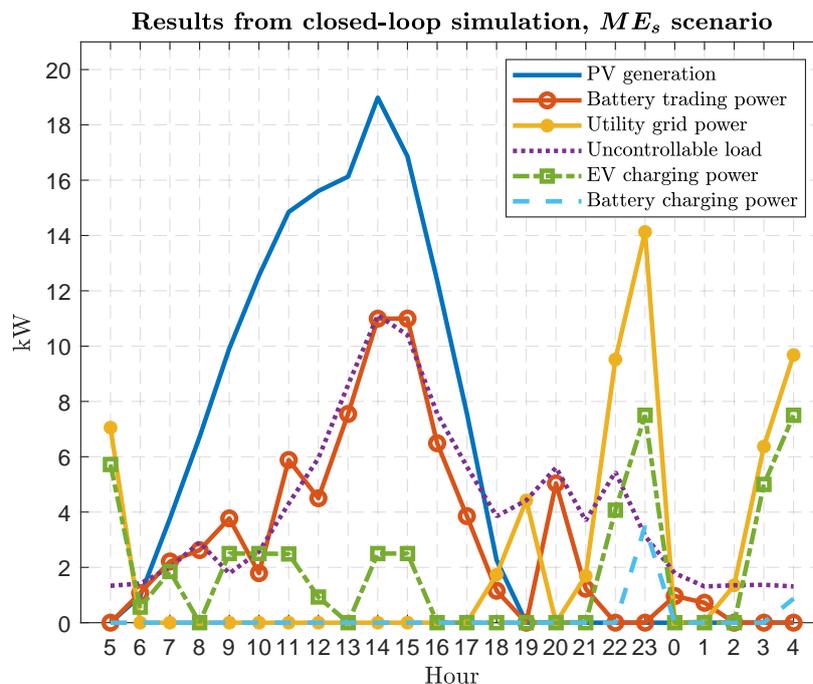


Figura 6.7 Resultados de la comunidad energética para el escenario ME_s .

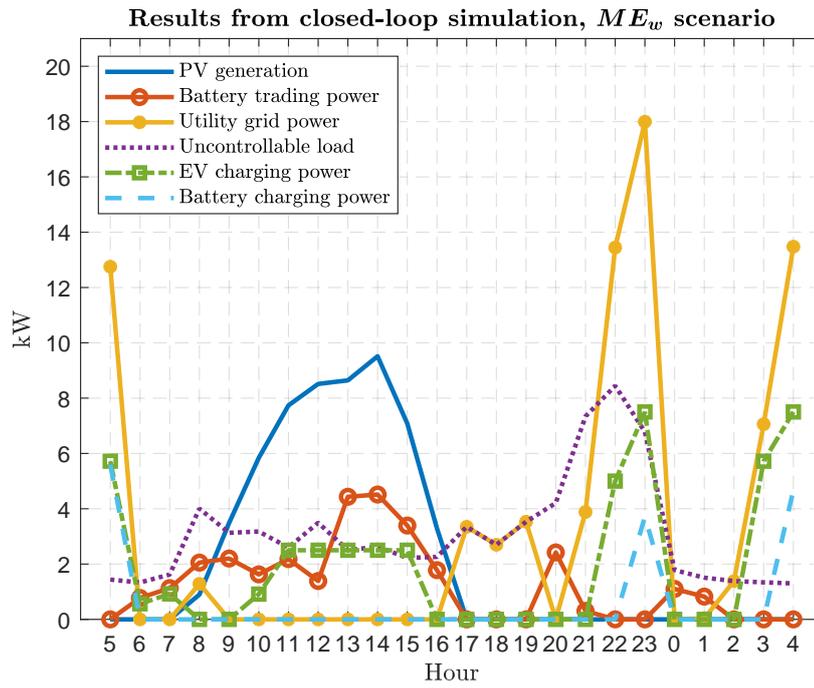


Figura 6.8 Resultados de la comunidad energética para el escenario ME_w .

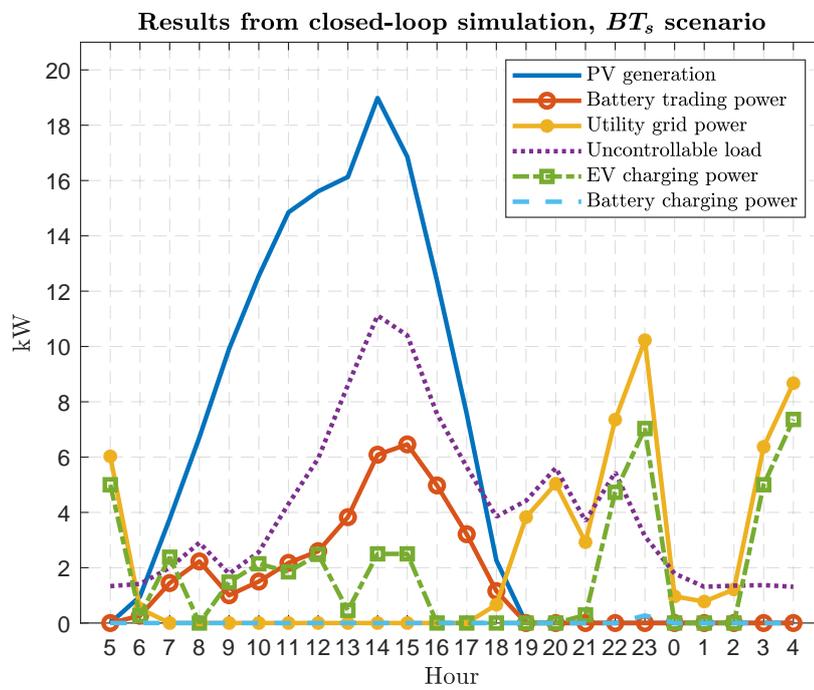


Figura 6.9 Resultados de la comunidad energética para el escenario BT_s .

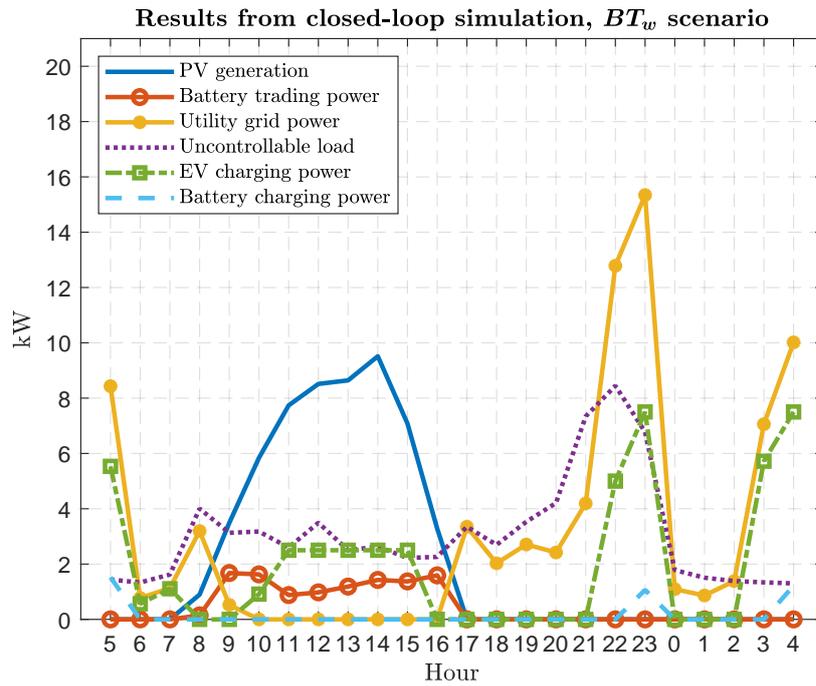


Figura 6.10 Resultados de la comunidad energética para el escenario BT_w .

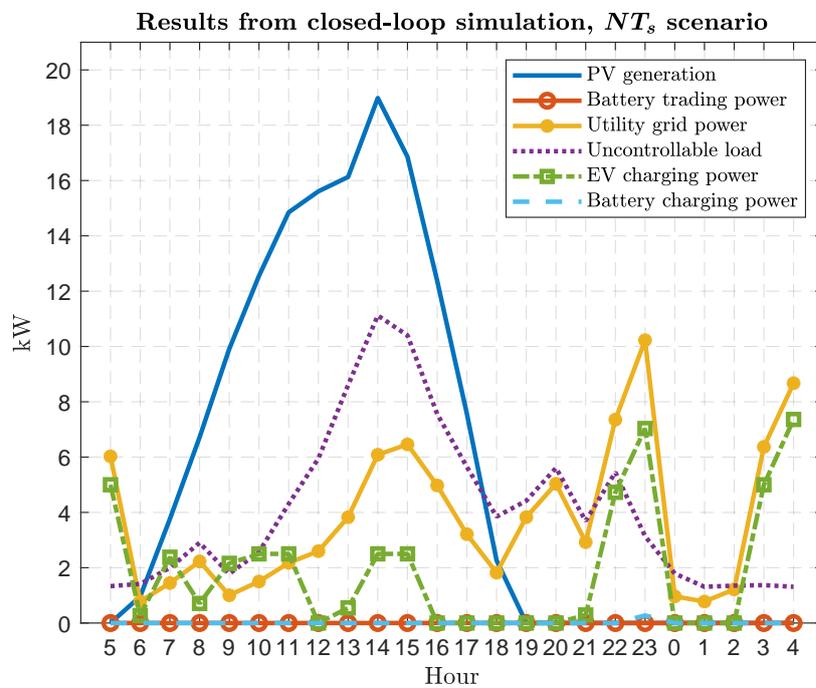


Figura 6.11 Resultados de la comunidad energética para el escenario NT_s .

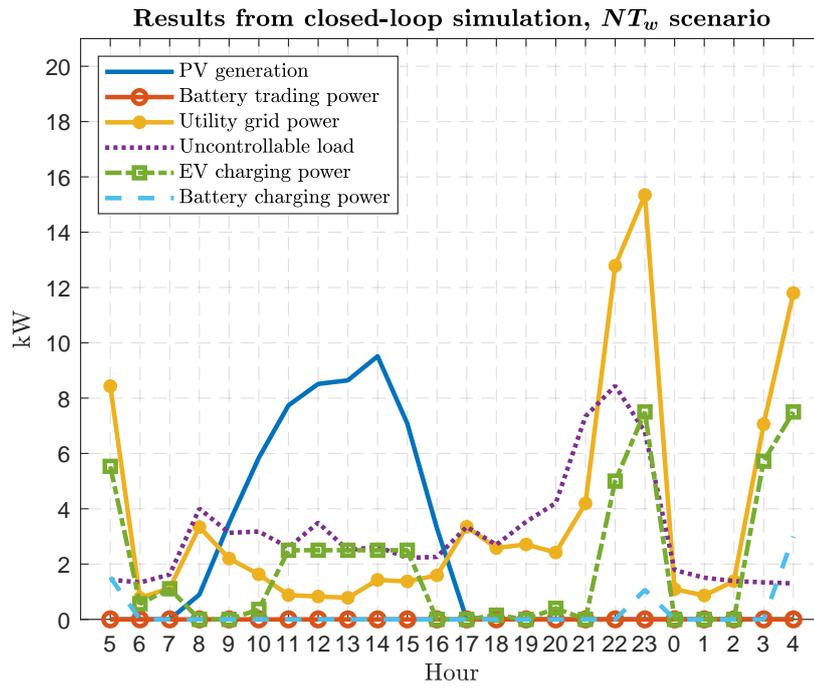


Figura 6.12 Resultados de la comunidad energética para el escenario NT_w .

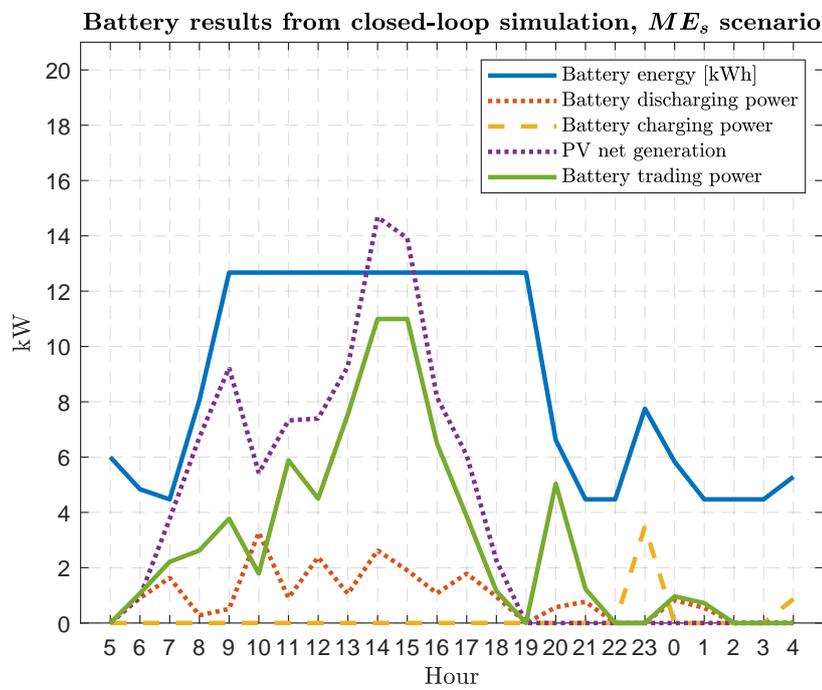


Figura 6.13 Resultados de las baterías para el escenario ME_s .

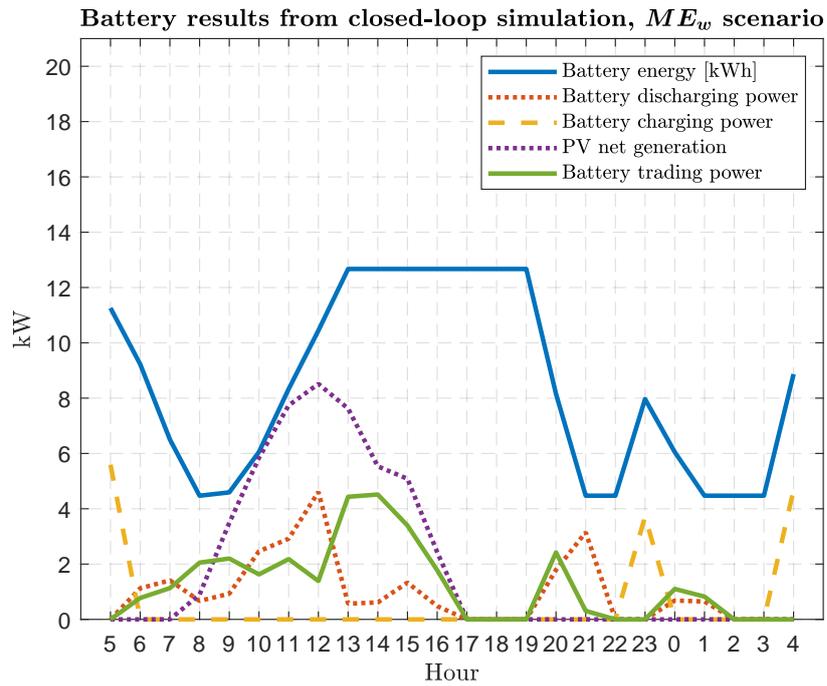


Figura 6.14 Resultados de las baterías para el escenario ME_w .

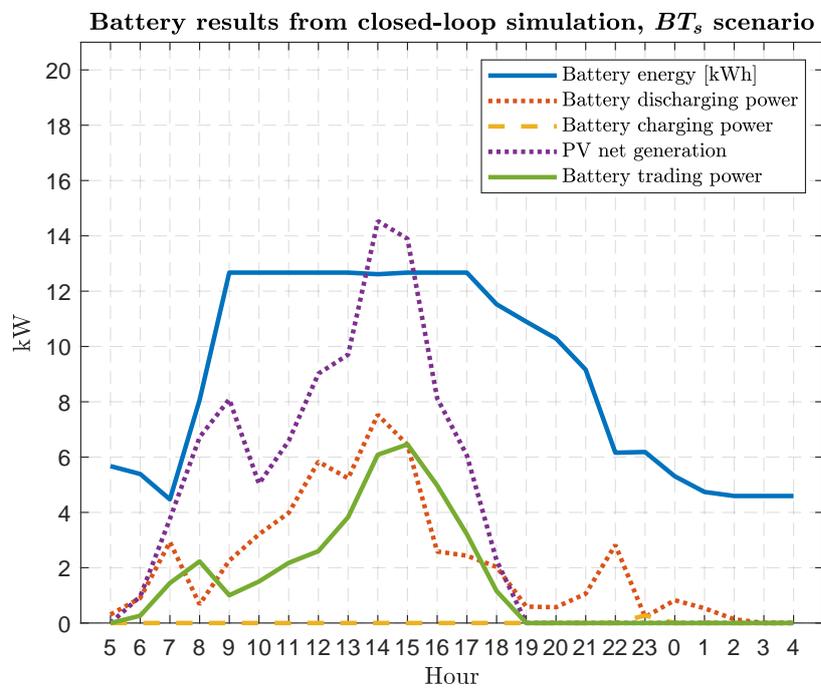


Figura 6.15 Resultados de las baterías para el escenario BT_s .

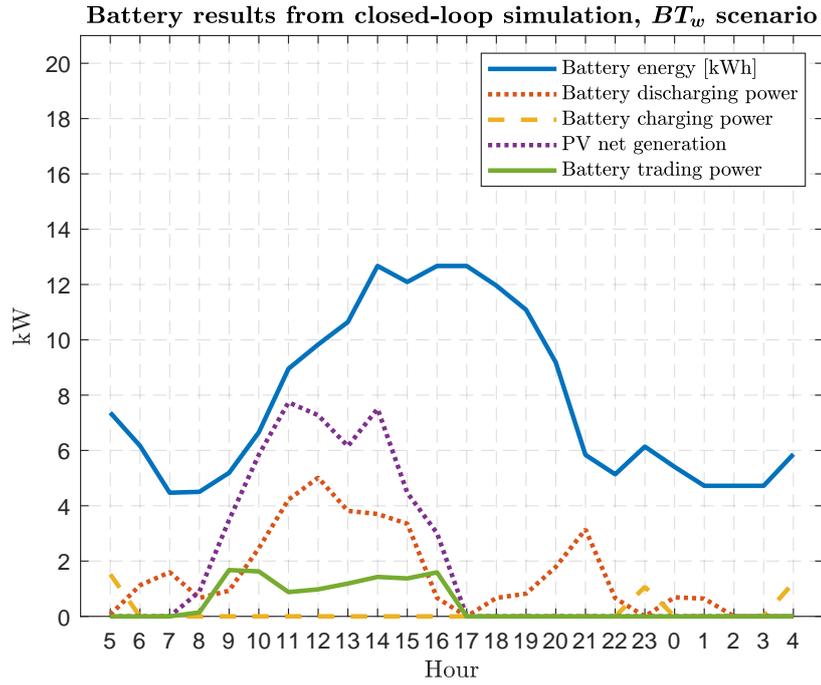


Figura 6.16 Resultados de las baterías para el escenario BT_w .

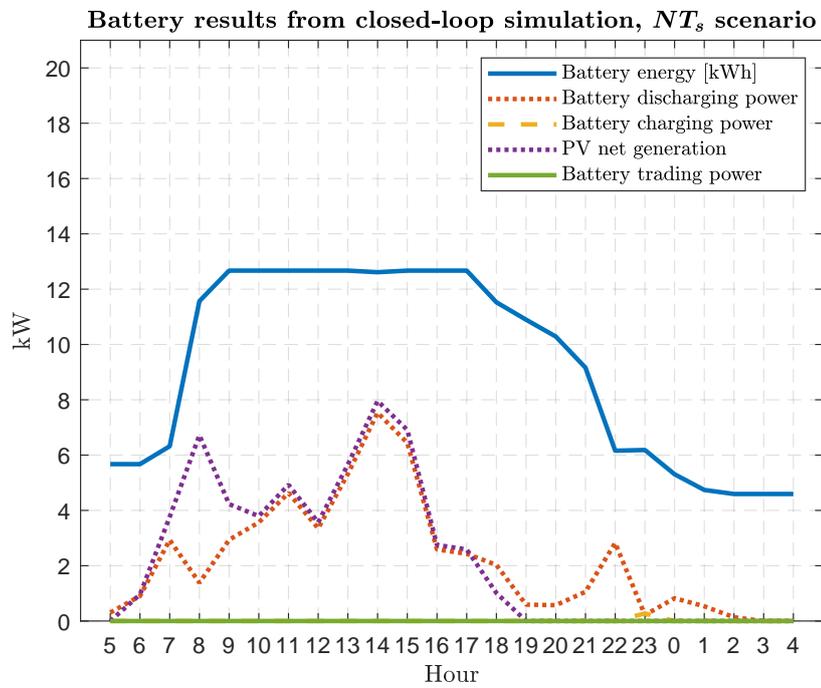


Figura 6.17 Resultados de las baterías para el escenario NT_s .

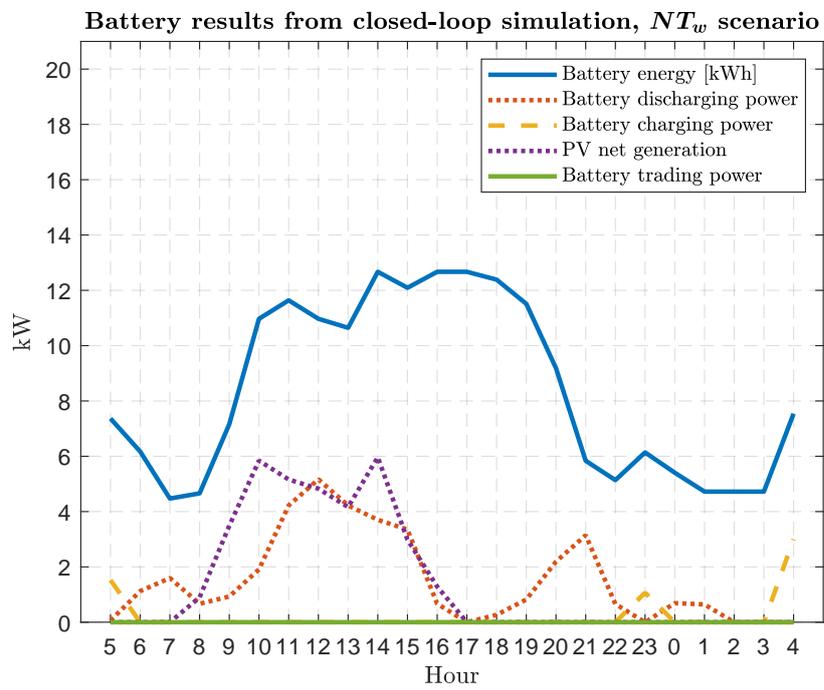


Figura 6.18 Resultados de las baterías para el escenario NT_w .

Tabla 6.1 Economic KPIs.

Economic KPIs	Scenarios					
	Summer			Winter		
	ME _s	BT _s	NT _s	ME _w	BT _w	NT _w
Prosumers						
Mean import costs (€)	3.19	2.42	2.42	4.74	3.78	4.03
Total import costs (€)	12.77	9.66	9.66	18.94	15.13	16.10
Mean trade costs (€)	0	0	0	0	0.06	0
Total trade costs (€)	0	9.63	0	0	0.23	0
Mean trade profits (€)	0	1.57	0	0	0.49	0
Total trade profits (€)	0	6.27	0	0	1.98	0
Mean costs (€)	3.19	2.42	2.42	4.74	3.84	4.03
Total costs (€)	12.77	9.66	9.66	18.94	15.35	16.10
Consumers						
Mean import costs (€)	0.88	1.47	3.56	1.19	1.76	2.34
Total import costs (€)	5.30	8.80	21.35	7.12	10.53	14.03
Mean trade costs (€)	0	1.05	0	0	0.29	0
Total trade costs (€)	0	6.27	0	0	1.75	0
Mean trade profits (€)	0	0	0	0	0	0
Total trade profits (€)	0	0	0	0	0	0
Mean costs (€)	0.88	2.51	3.56	1.19	2.047	2.34
Total costs (€)	5.30	15.08	21.35	7.12	12.28	14.03
Energy community						
Total import costs (€)	18.07	18.46	31.01	26.06	25.66	30.13
Total trade costs (€)	0	6.27	0	0	1.98	0
Total energy community costs (€)	18.07	24.73	31.01	26.06	27.64	30.13

Tabla 6.2 Power KPIs.

Energy usage KPIs	Scenarios					
	Summer			Winter		
	ME _s	BT _s	NT _s	ME _w	BT _w	NT _w
K ₁ (kWh)	70.84	36.93	0	30.12	10.88	0
K ₂ (kWh)	56.00	54.58	91.51	80.89	77.30	89.96
K ₃ (kWh)	22.01	53.11	53.11	23.38	36.03	36.03
K ₄ (%)	68.70	68.40	39.58	85.73	84.32	62.94

7 Conclusiones

Este trabajo ha presentado el desarrollo de una plataforma energética diseñada específicamente para comunidades de energía que están sujetas a perturbaciones estocásticas. Para gestionar estas incertidumbres, hemos implementado un esquema de control TB-MPC distribuido mediante el uso del algoritmo FC-MPC. Un aspecto destacado de este enfoque es la utilización de la tecnología blockchain, en particular, el uso de contratos inteligentes en la red de Ethereum. Esto ha permitido prescindir de terceras partes para coordinar y gestionar el algoritmo, lo que da como resultado un algoritmo completamente distribuido.

Los resultados obtenidos en este estudio demuestran la eficiencia y eficacia de esta plataforma. La tecnología blockchain ha desempeñado un papel fundamental al eliminar la necesidad de intermediarios y proporcionar transparencia y seguridad en el proceso de comercio de energía. El esquema jerárquico implementado ha demostrado que no se produce pérdida de descentralización, ya que los contratos inteligentes desempeñan un papel coordinador en la capa superior sin depender de entidades externas. También se ha prestado una atención especial a la automatización del flujo de control del algoritmo desplegado en blockchain a través de eventos, lo que contribuye a la eficiencia y capacidad de respuesta del sistema en tiempo real. El uso de una formulación estocástica basada en árboles ha permitido mitigar el impacto de los escenarios extremos poco probables en la fase de control, sin descartar su consideración.

Con respecto al rendimiento económico global de la comunidad energética, se ha observado una reducción sustancial en todas las configuraciones de comercio en comparación con la configuración NT. Es importante señalar que en este estudio no se han considerado la velocidad de ejecución de las transacciones en Ethereum, ni las tarifas asociadas, ya que la tecnología blockchain es un campo en constante evolución y son parámetros que dependen de múltiples factores.

Los trabajos futuros se centrarán en desarrollar algoritmos distribuidos aún más eficientes y capaces de hacer frente a agentes deshonestos. Por ejemplo, se explorarán estrategias para mitigar la influencia de hogares que optimizan un índice de rendimiento modificado para maximizar su propio beneficio en lugar del de la comunidad energética.

Índice de Figuras

3.1	Transacciones realizadas en Ethereum entre 2015 y 2023 [25]	12
3.2	Ethereum scalability trilemma [50]	16
4.1	Árbol de perturbaciones	22
4.2	Árbol de control	22
4.3	Algoritmo genérico FC-MPC [75]	23
5.1	Interfaz gráfica de usuario	32
5.2	Arquitectura de la aplicación desarrollada	33
5.3	Integración de Defender Services junto al contrato inteligente	34
6.1	Comunidad de energía simulada	38
6.2	Precio al que se compra la electricidad de la red de servicios públicos	38
6.3	Disponibilidad de carga de vehículos eléctricos. Las casillas amarillas y azules representan los intervalos de tiempo de carga de vehículos eléctricos disponibles y no disponibles, respectivamente	38
6.4	Evolución del árbol de perturbaciones a lo largo del horizonte de predicción	39
6.5	Conjunto de árboles de perturbación que pueden ser generados a las 0 AM	40
6.6	Generación total de energía solar fotovoltaica y demanda de carga fija en la comunidad de energía en verano e invierno	41
6.7	Resultados de la comunidad energética para el escenario ME_s	42
6.8	Resultados de la comunidad energética para el escenario ME_w	43
6.9	Resultados de la comunidad energética para el escenario BT_s	43
6.10	Resultados de la comunidad energética para el escenario BT_w	44
6.11	Resultados de la comunidad energética para el escenario NT_s	44
6.12	Resultados de la comunidad energética para el escenario NT_w	45
6.13	Resultados de las baterías para el escenario ME_s	45
6.14	Resultados de las baterías para el escenario ME_w	46
6.15	Resultados de las baterías para el escenario BT_s	46
6.16	Resultados de las baterías para el escenario BT_w	47
6.17	Resultados de las baterías para el escenario NT_s	47
6.18	Resultados de las baterías para el escenario NT_w	48

Índice de Tablas

5.1	Nomenclatura del modelo de la comunidad energética	26
6.1	Economic KPIs	49
6.2	Power KPIs	49

Índice de Códigos

Bibliografía

- [1] *Double-spending*, <https://en.wikipedia.org/wiki/Double-spending>, último acceso 15/11/22.
- [2] *Ethereum gas and fees*, <https://ethereum.org/en/developers/docs/gas/>, last accessed 2021/11/02.
- [3] *Ethereum networks*, <https://ethereum.org/en/developers/docs/networks/>, last accessed 2022/06/02.
- [4] *Ethereum virtual machine (evm)*, <https://ethereum.org/en/developers/docs/evm/>, last accessed 2021/10/26.
- [5] *Infura*, <https://docs.infura.io/infura/>, last accessed 2022/04/27.
- [6] *Metamask docs*, <https://docs.metamask.io/guide/>, last accessed 2022/06/27.
- [7] *Openzeppelin defender*, <https://docs.openzeppelin.com/defender/>, last accessed 2022/04/27.
- [8] *Photovoltaic geographical information system*, https://joint-research-centre.ec.europa.eu/pygis-photovoltaic-geographical-information-system_en, last accessed 2022/06/02.
- [9] *React*, <https://es.reactjs.org/>, last accessed 2022/04/27.
- [10] *Solidity*, <https://docs.soliditylang.org/en/develop/index.html>, last accessed 2022/04/27.
- [11] *Web3.js*, <https://web3js.readthedocs.io/en/v1.3.4/>, last accessed 2022/04/27.
- [12] Andreas M Antonopoulos, *Mastering bitcoin: unlocking digital cryptocurrencies*, " O'Reilly Media, Inc.", 2014.
- [13] Carlos Bordons, Félix Garcia-Torres, and Miguel A Ridao, *Model predictive control of microgrids*, vol. 358, Springer, 2020.
- [14] Tim Bray, *The javascript object notation (json) data interchange format*, Tech. report, 2014.
- [15] Vitalik Buterin et al., *Ethereum white paper*, GitHub repository **1** (2013), 22–23.
- [16] Eduardo F Camacho and Carlos Bordons Alba, *Model predictive control*, Springer science & business media, 2013.
- [17] Andrea Camisa and Giuseppe Notarstefano, *A distributed mixed-integer framework to stochastic optimal microgrid control*, IEEE Transactions on Control Systems Technology (2022).
- [18] Coin Market Cap, *Ethereum market cap*, <https://coinmarketcap.com/currencies/ethereum/>, last accessed 2023/11/05.
- [19] Usman W Chohan, *The double spending problem and cryptocurrencies*, Available at SSRN 3090174 (2021).
- [20] Stefano Raimondi Cominesi, Marcello Farina, Luca Giulioni, Bruno Picasso, and Riccardo Scattolini, *A two-layer stochastic model predictive control scheme for microgrids*, IEEE Transactions on Control Systems Technology **26** (2017), no. 1, 1–13.

- [21] Massimo Di Pierro, *What is the blockchain?*, Computing in Science & Engineering **19** (2017), no. 5, 92–95.
- [22] Whitfield Diffie and Martin E Hellman, *Multiuser cryptographic techniques*, Proceedings of the June 7-10, 1976, national computer conference and exposition, 1976, pp. 109–112.
- [23] European environment agency, *Share of energy consumption from renewable sources in europe*, <https://www.eea.europa.eu/ims/share-of-energy-consumption-from#:~:text=The%20EU%20had%20set%20the,2019%20to%2022.1%25%20in%202020.>, last accessed 2022/04/27.
- [24] Ayman Esmat, Martijn de Vos, Yashar Ghiassi-Farrokhfal, Peter Palensky, and Dick Epema, *A novel decentralized platform for peer-to-peer energy trading market with blockchain technology*, Applied Energy **282** (2021), 116123.
- [25] Etherscan.io, *Ethereum daily transactions chart*, <https://etherscan.io/chart/tx>, last accessed 2023/11/07.
- [26] Hal Finney, *Reusable proof of work*, https://www.wikiwand.com/es/Reusable_Proof_Of_Work, last accessed 2023/10/22.
- [27] Tyler Formica and Michael Pecht, *Return on investment analysis and simulation of a 9.12 kilowatt (kw) solar photovoltaic system*, Solar Energy **144** (2017), 629–634.
- [28] Felix Garcia-Torres, Carlos Bordons, and Miguel A Ridaó, *Optimal economic schedule for a network of microgrids with hybrid energy storage system using distributed model predictive control*, IEEE transactions on industrial electronics **66** (2018), no. 3, 1919–1929.
- [29] Juan M Grosso, Pablo Velarde, Carlos Ocampo-Martinez, Jose María Maestre, and Vicenç Puig, *Stochastic model predictive control approaches applied to drinking water networks*, Optimal Control Applications and Methods **38** (2017), no. 4, 541–558.
- [30] Stuart Haber and W Stornetta, *How to time-stamp a digital document, crypto'90, Incs 537*, 1991.
- [31] Christian A Hans, Pantelis Sopasakis, Jörg Raisch, Carsten Reincke-Collon, and Panagiotis Patrinos, *Risk-averse model predictive operation control of islanded microgrids*, IEEE transactions on control systems technology **28** (2019), no. 6, 2136–2151.
- [32] Ali Hooshmand, Mohammad H Poursaeidi, Javad Mohammadpour, Heidar A Malki, and Karolos Grigoriadis, *Stochastic model predictive control method for microgrid management*, 2012 IEEE PES Innovative Smart Grid Technologies (ISGT), IEEE, 2012, pp. 1–7.
- [33] IEA, *Renewables 2021*, <https://www.iea.org/reports/renewables-2021>, last accessed 2022/06/17.
- [34] Mark Z Jacobson and Mark A Delucchi, *Providing all global energy with wind, water, and solar power, part i: Technologies, energy resources, quantities and areas of infrastructure, and materials*, Energy policy **39** (2011), no. 3, 1154–1169.
- [35] Craig Jarvis, *Cypherpunk ideology: objectives, profiles, and influences (1992–1998)*, Internet Histories **6** (2022), no. 3, 315–342.
- [36] Jaynti Kanani, Sandeep Nailwal, and Anurag Arjun, *Matic whitepaper*, Polygon, Bengaluru, India, Tech. Rep., Sep (2021).
- [37] Rabiya Khalid, Nadeem Javaid, Ahmad Almogren, Muhammad Umar Javed, Sakeena Javaid, and Mansour Zuair, *A blockchain-based load balancing in decentralized hybrid p2p energy trading market in smart grid*, IEEE Access **8** (2020), 47047–47062.
- [38] Jose María Maestre, L Raso, PJ Van Overloop, and B De Schutter, *Distributed tree-based model predictive control on a drainage water system*, Journal of Hydroinformatics **15** (2013), no. 2, 335–347.
- [39] Valérie Masson-Delmotte, Panmao Zhai, Hans-Otto Pörtner, Debra Roberts, Jim Skea, Priyadarshi R Shukla, et al., *Global warming of 1.5 c: Ipcc special report on impacts of global warming of 1.5 c above pre-industrial levels in context of strengthening response to climate change, sustainable development, and efforts to eradicate poverty*, Cambridge University Press, 2022.

- [40] Esther Mengelkamp, Johannes Gärtner, Kerstin Rock, Scott Kessler, Lawrence Orsini, and Christof Weinhardt, *Designing microgrid energy markets: A case study: The brooklyn microgrid*, Applied Energy **210** (2018), 870–880.
- [41] Ali Mesbah, *Stochastic model predictive control: An overview and perspectives for future research*, IEEE Control Systems Magazine **36** (2016), no. 6, 30–44.
- [42] Thomas Morstyn and Malcolm D McCulloch, *Multiclass energy management for peer-to-peer energy trading driven by prosumer preferences*, IEEE Transactions on Power Systems **34** (2018), no. 5, 4005–4014.
- [43] John M Mulvey and Andrzej Ruszczyński, *A new scenario decomposition method for large-scale stochastic optimization*, Operations research **43** (1995), no. 3, 477–490.
- [44] Hannah E Murdock, Duncan Gibb, Thomas André, Janet L Sawin, Adam Brown, Lea Ranalder, Ute Collier, Christopher Dent, Baerbel Epp, Chetna Hareesh Kumar, et al., *Renewables 2021-global status report*, (2021).
- [45] David Muñoz de la Peña, Alberto Bemporad, and Teodoro Alamo, *Stochastic programming applied to model predictive control*, Proceedings of the 44th IEEE Conference on Decision and Control, IEEE, 2005, pp. 1361–1366.
- [46] Satoshi Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, Bitcoin.—URL: <https://bitcoin.org/bitcoin.pdf> **4** (2008).
- [47] ———, *Bitcoin: A peer-to-peer electronic cash system*, Decentralized Business Review (2008), 21260.
- [48] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder, *Bitcoin and cryptocurrency technologies: a comprehensive introduction*, Princeton University Press, 2016.
- [49] Bruce Jay Nelson, *Remote procedure call*, Carnegie Mellon University, 1981.
- [50] Ethereum Organization, *The challenge of decentralized scaling*, <https://ethereum.org/en/roadmap/vision/#:~:text=Ethereum%20needs%20to%20reduce%20network%20congestion%20and%20improve,problems%20would%20be%20to%20make%20it%20more%20centralized.>, last accessed 2023/11/06.
- [51] ———, *Danksharding*, <https://ethereum.org/en/roadmap/danksharding/>, last accessed 2023/11/09.
- [52] Ethereum organization, *Ethereum history*, <https://ethereum.org/es/history/>, last accessed 2023/11/06.
- [53] Ethereum Organization, *Ethereum layer 2 scaling*, <https://ethereum.org/en/developers/docs/scaling/#layer-2-scaling>, last accessed 2023/11/09.
- [54] ———, *Ethereum scaling*, <https://ethereum.org/en/developers/docs/scaling/>, last accessed 2023/11/09.
- [55] Ethereum organization, *History of ethereum*, <https://ethereum.org/en/history/>, last accessed 2023/11/05.
- [56] Ethereum Organization, *How the merge impacted eth supply*, <https://ethereum.org/en/roadmap/merge/issuance/>, last accessed 2023/11/06.
- [57] ———, *Merkle patricia trie*, <https://ethereum.org/en/developers/docs/data-structures-and-encoding/patricia-merkle-trie/>, last accessed 2023/11/06.
- [58] ———, *Optimistic rollups*, <https://ethereum.org/en/developers/docs/scaling/optimistic-rollups/>, last accessed 2023/11/09.
- [59] ———, *Zero-knowledge rollups*, <https://ethereum.org/en/developers/docs/scaling/zk-rollups/>, last accessed 2023/11/09.
- [60] Ethereum Classic Organization, *Ethereum classic*, <https://ethereumclassic.org/>, last accessed 2023/11/06.

- [61] EJ Palacios-Garcia, A Moreno-Munoz, I Santiago, JM Flores-Arias, FJ Bellido-Outeirino, and IM Moreno-Garcia, *A stochastic modelling and simulation approach to heating and cooling electricity consumption in the residential sector*, *Energy* **144** (2018), 1080–1091.
- [62] Alessandra Parisio, Evangelos Rikos, and Luigi Glielmo, *A model predictive control approach to microgrid operation optimization*, *IEEE Transactions on Control Systems Technology* **22** (2014), no. 5, 1813–1827.
- [63] Polkadot, *Polkadot network*, <https://polkadot.network/features/technology/>, last accessed 2023/11/05.
- [64] R Tyrrell Rockafellar and Roger J-B Wets, *Scenarios and policy aggregation in optimization under uncertainty*, *Mathematics of operations research* **16** (1991), no. 1, 119–147.
- [65] Arto Salomaa, *Public-key cryptography*, (1996).
- [66] David Siegel, *Understanding the dao attack*, <https://www.coindesk.com/learn/understanding-the-dao-attack/>, last accessed 2023/11/06.
- [67] Manuel Sivianes, José M Maestre, Ascensión Zafra-Cabeza, and Carlos Bordons, *Blockchain for energy trading in energy communities using stochastic and distributed model predictive control*, *IEEE Transactions on Control Systems Technology* (2023).
- [68] Manuel Sivianes Castaño, *Design of a blockchain-based platform for peer-to-peer energy trading*, (2021).
- [69] Rajeev Sobti and Ganesan Geetha, *Cryptographic hash functions: a review*, *International Journal of Computer Science Issues (IJCSI)* **9** (2012), no. 2, 461.
- [70] Cryptopedia Staff, *What was the dao?*, <https://www.gemini.com/cryptopedia/the-dao-hack-makerdao#section-what-is-a-dao>, last accessed 2023/11/06.
- [71] Nick Szabo, *Bit gold*, Recuperado de <https://nakamotoinstitute.org/bit-gold/TVer> página (2005).
- [72] Gijs van Leeuwen, Tarek AlSkaif, Madeleine Gibescu, and Wilfried van Sark, *An integrated blockchain-based energy management platform with bilateral trading for microgrid communities*, *Applied Energy* **263** (2020), 114613.
- [73] P Velarde, X Tian, AD Sadowska, and Jose María Maestre, *Scenario-based hierarchical and distributed mpc for water resources management with dynamical uncertainty*, *Water Resources Management* **33** (2019), no. 2, 677–696.
- [74] Pablo Velarde, L Valverde, Jose María Maestre, Carlos Ocampo-Martínez, and C Bordons, *On the comparison of stochastic model predictive control strategies applied to a hydrogen-based microgrid*, *Journal of Power Sources* **343** (2017), 161–173.
- [75] Aswin N Venkat, James B Rawlings, and Stephen J Wright, *Stability and optimality of distributed model predictive control*, *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE, 2005, pp. 6680–6685.
- [76] Boyu Wang, Morteza Dabbaghjamanesh, Abdollah Kavousi-Fard, and Shahab Mehraeen, *Cybersecurity enhancement of power trading within the networked microgrids based on blockchain and directed acyclic graph approach*, *IEEE Transactions on Industry Applications* **55** (2019), no. 6, 7300–7309.
- [77] Polkadot wiki, *Polkadot consensus*, <https://wiki.polkadot.network/docs/learn-consensus>, last accessed 2023/11/06.
- [78] Yan Zhang, Fanlin Meng, Rui Wang, Behzad Kazemtabrizi, and Jianmai Shi, *Uncertainty-resistant stochastic mpc approach for optimal operation of chp microgrid*, *Energy* **179** (2019), 1265–1278.
- [79] Yi Zheng, Shaoyuan Li, and Ruomu Tan, *Distributed model predictive control for on-connected microgrid power management*, *IEEE Transactions on Control Systems Technology* **26** (2017), no. 3, 1028–1039.

Índice alfabético
