



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

REDIRECCIÓN Y OPTIMIZACIÓN DEL CONSUMO ELÉCTRICO DE UNA  
MICRO-RED HACIENDO USO DEL PROTOCOLO DATA DISTRIBUTION SYSTEM

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL ELÉCTRICO

SERGIO ANDRÉS GALLARDO REBOLLEDO

PROFESOR GUÍA:  
CLAUDIO ESTÉVEZ MONTERO

MIEMBROS DE LA COMISIÓN:  
GUILLERMO JIMÉNEZ ESTÉVEZ  
PATRICIO MENDOZA ARAYA

SANTIAGO DE CHILE  
2019

RESUMEN DE LA MEMORIA PARA OPTAR  
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO  
POR: SERGIO ANDRÉS GALLARDO REBOLLEDO  
FECHA: 23/08/19  
PROF. GUÍA: CLAUDIO ESTÉVEZ MONTERO

## REDIRECCIÓN Y OPTIMIZACIÓN DEL CONSUMO ELÉCTRICO DE UNA MICRO-RED HACIENDO USO DEL PROTOCOLO DATA DISTRIBUTION SYSTEM

En la actualidad existe un aumento en el crecimiento del mercado de micro-redes en el mundo y también la capacidad de poder enviar y recibir información importante dentro de la misma. La capacidad de poder transmitir y enviar información de forma automática y en tiempo real presenta ser una ventaja a la hora de tomar decisiones en cualquier red, más aun si es que es una micro-red, las cuales son conocidas por no poder generar potencia de forma constante. Es por esto que surge la necesidad de poder desarrollar una prueba de concepto sobre como poder generar un sistema de envío de información para poder otorgar de un cierto grado de autonomismo a la propia red.

En esta tesis se emulará una micro-red haciendo uso de datos reales obtenidos en la localidad de Huatacondo, el protocolo de envío y recepción de información será Data Distribution Service (DDS) con el agregado de que la forma en que los elementos de la red se descubran será de manera distribuida, lo anterior para eliminar la necesidad de un dispositivo central. La red emulada hace uso de el *hardware Raspberry Pi* y una interconexión *Wireless Local Area Network* (WLAN) para realizar la conexión física entre los componentes de la red emulada. Una vez exista la correcta transmisión de información, se propone ocupar los datos de potencia del generador diesel existente en Huatacondo para poder tener una referencia de como la red responde haciendo uso de la red de comunicaciones.

El sistema logra generar comunicación y traspaso de información entre los componentes, estimando de forma correcta la potencia necesaria para suplir la red, potencia que proviene del generador diesel, en los distintos intervalos de tiempo en la que la red opera. Por último se proponen distintas formas de mejorar la red de comunicaciones para una mejor implementación de la misma, en particular, se propone el acceso remoto de los distintos dispositivos para eliminar la necesidad de WLAN.



*Se lo dedico a mi familia.*



# Agradecimientos

Quiero agradecer a mi familia, amigos y comisión examinadora. Todos han sido importantes para la resolución de esta tesis.



# Tabla de Contenido

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Problemática . . . . .	2
1.3. Hipótesis . . . . .	3
1.4. Objetivo general . . . . .	3
1.5. Objetivos específicos . . . . .	4
1.6. Alcance . . . . .	4
1.7. Capítulos del documento . . . . .	5
<b>2. Marco teórico</b>	<b>6</b>
2.1. Distribución energética actual . . . . .	6
2.2. Micro-red . . . . .	7
2.2.1. Componentes en una micro-red . . . . .	8
2.3. <i>Data Distribution Service</i> . . . . .	10
2.3.1. Real-Time Publish-Subscribe . . . . .	12
2.3.2. Implementaciones de DDS . . . . .	12
2.4. <i>Hardware</i> . . . . .	14
2.4.1. Raspberry Pi . . . . .	14
<b>3. Metodología</b>	<b>15</b>
3.1. Datos para la emulación . . . . .	15
3.1.1. PV . . . . .	15
3.1.2. Demanda . . . . .	16
3.1.3. Baterías . . . . .	17
3.1.4. Genset . . . . .	17
3.1.5. Conclusión de los datos analizados . . . . .	18
3.2. Arquitectura de la red . . . . .	18
3.2.1. Descubrimiento descentralizado . . . . .	21
<b>4. Resultados</b>	<b>22</b>
4.0.1. Análisis . . . . .	22
4.0.2. Errores aleatorios de conexión . . . . .	23
<b>Conclusión</b>	<b>23</b>
4.0.3. Trabajo futuro . . . . .	25
<b>Bibliografía</b>	<b>27</b>

<b>Anexo A</b>	<b>28</b>
4.1. Sobre el terminal de las distribuciones GNU-Linux . . . . .	29
4.1.1. Sobre <i>sudo</i> . . . . .	29
4.1.2. Comando útiles . . . . .	29
4.2. Instalación Raspbian . . . . .	31
4.3. Instalación de OpenDDS . . . . .	32
4.3.1. Descarga y compilación de OpenDDS . . . . .	32
4.3.2. Conexión SSH . . . . .	33
4.3.3. Traspaso de OpenDDS a la Raspberry Pi . . . . .	34
4.3.4. Ejemplo de instalación . . . . .	35
4.4. Sobre el uso de OpenDDS . . . . .	35

# Índice de Ilustraciones

2.1. Micro-red básica. Notar que la red de distribución nacional fue incorporada solo por completitud y no será considerada un componente existente para el desarrollo de la micro-red a emular. Imagen editada de [1] . . . . .	9
2.2. Arquitectura básica de cómo opera una red de información en base a DDS. . . . .	11
2.3. Logo de <i>OpenDDS</i> . . . . .	13
2.4. Raspberry Pi Modelo 3B. . . . .	14
3.1. La localidad de Huatacondo. . . . .	16
3.2. Potencia de generación fotovoltaica de la red. . . . .	16
3.3. Perfil de la demanda de la red, superpuesta con la potencia entregada por los paneles fotovoltaicos. . . . .	17
3.4. Potencia entregada y consumida de las baterías en la red. . . . .	17
3.5. Potencia entregada por el generador diesel superpuesta con la potencia generada de los paneles fotovoltaicos. . . . .	18
3.6. Potencia combinada de todos los elementos de la red. . . . .	19
3.7. Comparación entre la micro-red real y la emulación propuesta. . . . .	20
4.1. Datos recibidos y desplegados por el lector de datos. Caso cuando el generador si se encuentra activo en la red. Los datos representan valores en [kW] . . . . .	23
4.2. Datos recibidos y desplegados por el lector de datos. Caso cuando el generador si se encuentra activo en la red. Los datos representan valores en [kW] . . . . .	24



# Capítulo 1

## Introducción

Este trabajo comienza mostrando la motivación con la que se basa este documento, una descripción de la problemática existente y los objetivos tanto generales como particulares con los que se planea hacer frente a la problemática planteada. Como este trabajo se enfoca directamente a las micro-redes es que se hará necesaria una contextualización general de este tema en particular tanto en Chile como en el mundo.

### 1.1. Motivación

En la actualidad existe un mayor interés en la implementación y operación de sistemas autónomos<sup>1</sup> de generación y distribución energética, más conocidos como micro-redes. Las micro-redes se alejan de métodos de generación y distribución energética tradicionales debido a que se hace enfoque en la sustentabilidad energética de las propias celdas; estas siendo sistemas energéticos locales<sup>2</sup> que tienen elementos de generación, distribución, almacenamiento y consumo en la propia red. Aunque no es necesario en un principio, muchos sistemas presentan elementos de generación que caen dentro de la categoría de *Energías Renovables No Convencionales* (ERNC).

Debido a este interés emergente en Chile de las ERNC y en específico, las micro-redes, es que a nivel de universidad se han generado proyectos y laboratorios[2] para poder seguir desarrollando esta tecnología. Las micro-redes son sistemas donde su principal objetivo es la generación, distribución, almacenamiento y consumos de energía eléctrica de manera autónoma pero un sistema así puede verse altamente beneficiado por el uso de tecnologías de telecomunicaciones para así saber el estado eléctrico de la red en todo momento. De esta forma, se pretende investigar e implementar un sistema de telecomunicación perteneciente a la propia micro-red y así estudiar su comportamiento bajo este contexto.

---

<sup>1</sup>También pueden caer en esta categoría sistemas semi-autónomos donde puede existir operación humana directa hasta un cierto grado.

<sup>2</sup>La definición de *localidad* no es única pero se da a entender que es un sistema no tan extenso como los existentes en países o regiones a través del mundo.

Además de las consideraciones energéticas también existe una motivación netamente tele-comunicacional: la capacidad de alcanzar intercambio de información sin la necesidad de un agente central o servidor para llevar a cabo la tarea. Como las micro-redes son entidades donde los agentes dentro de la misma se encuentran distribuidos dentro de la misma red, se hace más evidente la necesidad de descentralizar la generación y procesamiento de información. De esta forma se espera que la falla de cualquier elemento de la red de telecomunicaciones no genere una caída completa del sistema. Junto con implementarse un sistema de intercambio de información embebida en la red de generación, resulta útil poder generar documentación para el posterior uso o desarrollo de la red de telecomunicaciones, permitiendo acelerar el desarrollo de esta área.

## 1.2. Problemática

En el presente, las micro-redes pueden hacer uso extensivos de generadores tipo ERNC, el problema de ocupar este tipo de tecnologías es la poca constancia en la capacidad de generación de potencia eléctrica a través del tiempo; variables como luminosidad solar en el caso de paneles solares fotovoltaicos o magnitud y dirección del viento en el caso de aerogeneradores no permiten tener una estimación precisa de como se va a comportar la red eléctrica. La posibilidad de ocupar tecnologías de telecomunicación enfocado a micro-redes, abre un abanico de posibilidades en relación a como se puede mejorar el rendimiento a nivel energético de una MR. Comunicación entre elementos de la MR no permite saber el estado actual de la misma en tiempo real; permitiendo la incorporación de un concepto de interés para los propósitos de este trabajo: La auto-regulación.

Una micro-red que tenga la capacidad de auto-regulación, es decir, la capacidad de tomar decisiones de forma autónoma en materia energética para la mejora de rendimiento de la misma es un concepto digno de investigarse pero trae consigo desafíos en materia de implementación y operación. En este sistema se observan distintos retos de implementación, tales como:

- Sensado.
- Comunicación.
- Inteligencia.

Donde cada uno de estas características debe de ser coordinada con las demás para generar un sistema coherente para la toma de decisiones.

El sensado se refiere a la adquisición de datos en tiempo real para su posterior uso. Un parámetro como potencia sirven para poder caracterizar el sistema en todo momento. Vale mencionar que este trabajo busca poder emular todos los elementos de la red, generadores<sup>3</sup>, almacenadores<sup>4</sup> y consumidores<sup>5</sup>. Para comunicar los componentes y permitir el libre traspaso

---

<sup>3</sup>paneles solares o aerogeneradores

<sup>4</sup>baterías

<sup>5</sup>domicilios o recintos industriales

de información es que se debe ocupar un protocolo de comunicación pertinente y efectivo. Debido a que se planea enviar la información eléctrica de un elemento de la red a todos los demás elementos es que se considera ocupar un protocolo publicador-subscriptor capaz de generar el intercambio de información antes mencionado. Se verá en este trabajo que el protocolo a ocupar es *Data Distribution Service* (DDS). Por último, se necesita de un algoritmo que considere los estados de todos los elementos de la red en tiempo real y de forma automática la red, mediante actuadores, es capaz de redirigir la energía eléctrica tal que se produzca un aumento neto en el rendimiento de la misma.

### 1.3. Hipótesis

Se estima entonces que debe existir un protocolo de transmisión de datos entre elementos de una red que pueda satisfacer necesidades como confiabilidad, seguridad y eficacia a la hora de manejar el volumen de datos generados de forma periódica. Junto con esto, también es necesario algún cierto grado de autonomía y toma de decisiones de la red ante la existencia de escasez de potencia en la misma.

Un protocolo existente, el cual es basado en el paradigma publicador y subscriptor permite modularidad, seguridad y compatibilidad con aplicaciones que puede implementar el usuario para poder manejar y controlar la red. Se propone entonces que el protocolo DDS permite la implementación de una red de comunicaciones paralela a la de potencia con la capacidad de envío de información y autoregulación con la mínima participación humana.

### 1.4. Objetivo general

Se requiere implementar un sistema de emulación de sensado, comunicaciones e inteligencia computacional capaz de mejorar el rendimiento eléctrico de la red. El sistema debe ser autónomo o por lo menos requerir de la menor cantidad de interacción humana posible para su correcto funcionamiento. Los datos a ocupar deben estar siendo procesados en tiempo real y todos los elementos de la red deben estar correctamente conectados.

Es de importancia poder generar una red de comunicaciones descentralizada, capaz de eliminar la necesidad de un servidor central donde se concentre la adquisición y procesamiento de la información de la red. Se propone también generar una plataforma base para el posterior desarrollo de esta tecnología a cualquier investigador interesado en esta área.

## 1.5. Objetivos específicos

- Crear una red para pruebas de concepto<sup>6</sup>.
- Emular datos adquiridos por un sensor para cada elemento de la red.
- Investigar e implementar algún protocolo de comunicación basado en el intercambio de información.
- Investigar y crear un algoritmo de toma de decisiones el cual será responsable de mejorar el rendimiento de la red mediante actuadores.
- Diseñar red de comunicaciones a implementar con una arquitectura descentralizada.
- Documentar de forma exhaustiva y explicar la parte técnica del proyecto realizado.

## 1.6. Alcance

El foco de interés de este trabajo es la implementación de todos los componente mencionados en 1.5 para su posterior prueba y análisis. Un trabajo así se puede extender a más actividades como:

- Creación de librerías en un único lenguaje de programación.
- Uso de componentes reales de una micro-red.
- Implementación de distintos tipos de algoritmos de toma de decisiones<sup>7</sup>.

Los puntos anteriores van a ser posibles mejoras al sistema que se presentará durante el documento pero no suponen lo central de este trabajo en particular. Lo anterior es debido a que se considera de suma importancia el prototipaje de un sistema de esta características, es decir, la correcta implementación, operación, documentación y por último prueba de concepto del sistema en su totalidad sin hacer a priori mejoras substanciales de los límites a lo que este sistema puede soportar.

Para comprobar el correcto funcionamiento e impacto del nuevo sistema de telecomunicaciones en la micro-red, es necesario usar un parámetro de interés, se considera la potencia efectiva suministrada a los distintos consumidores dentro de la red el parámetro suficiente para corroborar si existe un impacto evidente desarrollado en este trabajo. La prueba en particular que se ocupará es una principalmente de concepto donde se emulará tres componentes

---

<sup>6</sup>Esto quiere decir que la red a ocupar no necesariamente debe de ocupar elementos reales de una micro-red, sino que es posible ocupar datos de generación, consumo y almacenamiento ya existentes

<sup>7</sup>En este caso se entraría más a lo que es la disciplina de *Inteligencia Computacional* donde existen métodos como redes neuronales, lógica difusa, *support vector machine*, entre otros. Todos estos métodos permiten que sistemas genéricos logren la capacidad de tomar distintas decisiones dependiendo del entrenamiento entregados y la información existente actual.

representativos de una red, un elemento generador, un elemento almacenador y por último un elemento consumidor, así se podrá tener una primera noción de como puede operar y mejorar este sistema en un ambiente real, con componentes reales.

## 1.7. Capítulos del documento

Para claridad del trabajo desarrollado, este se divide en cinco capítulos distintos, un capítulo de Introducción para mencionar los distintos objetivos a cumplir que justifican el presente documento, además de detallar los alcances y motivaciones existentes. Marco Teórico donde se explican y definen los distintos elementos necesarios para poder entender el desarrollo de esta tesis, con descripciones de las distintas tecnologías a ocupar. Explicados los distintos elementos que formarán parte de este trabajo, se procede a la Metodología, el cual es el capítulo del cómo se implementan los distintos sistemas de interés, también justificando supuestos y tecnicismos necesarios para obtener resultados importantes. El capítulo de Resultados presenta los distintos resultados obtenidos implementando el sistema propuesto en el capítulo anterior, con un breve análisis del porqué se obtuvieron en primer lugar. Por último, las Conclusiones muestran como los objetivos principales y generales de este trabajo se cumplen además de presentar una propuesta de trabajo futuro posible con este documento.

# Capítulo 2

## Marco teórico

En esta sección se dará contexto y explicación a todos los elementos importantes para el desarrollo de este trabajo, desde tecnologías a utilizar hasta consideraciones eléctricas a nivel país. La forma en que se presentará las distintas secciones en este capítulo será comenzando con definiciones más generales y luego avanzando a especificaciones más precisas de los componentes tecnológicos a ocupar.

### 2.1. Distribución energética actual

En Chile, para la generación, transmisión y distribución (GTD<sup>1</sup>) de energía eléctrica existen cuatro sistemas eléctricos importantes:

- Sistema Interconectado Central (SIC).
- Sistema Interconectado del Norte Grande (SING)<sup>2</sup>.
- Sistema Eléctrico de Aysen (SEA).
- Sistema Eléctrico de Magallanes (SEM).

La existencia de todos estos sistemas se deben principalmente a las características geográficas del país, donde los costos en integración y luego transmisión hacen difícil que se puedan hacer posibles estas opciones. Aún así, estos sistemas pueden ser considerados estándar, en el sentido de que sus métodos para entregar energía a todos los sectores de la sociedad no distan de otros métodos a través del mundo. En resumen, este método consiste principalmente en generación eléctrica distribuida en distintos puntos del territorio nacional donde

---

<sup>1</sup>Debido a que estas tres actividades son las más importantes para estos sistemas y serán mencionadas de manera reiterada en este documento, es que se referirá a estas mediante el presente acrónimo de ahora en adelante.

<sup>2</sup>El día 21 de noviembre de 2017 se concretó la interconexión entre el SING y el SIC[3], convirtiendo a ambos sistemas en términos prácticos en uno. En este documento se lista a ambos como distintos debido a razones históricas.

la propia generación sea posible<sup>3</sup>, transmisión de alto voltaje de dicha energía generada a centros urbanos o industriales y luego distribución a los distintos elementos pertenecientes a la red.

Para Chile, estos sistemas han permitido a través del tiempo mejorar el estándar de vida de las personas al entregar energía eléctrica necesaria para el diario vivir, con relativamente pocos cortes de energía<sup>4</sup> y alcanzando casi la totalidad de la población chilena. Aún así considerando los claros beneficios de un sistema (o sistemas) con estas características, existen ciertas desventajas que hacen imposible o por lo menos extremadamente costoso suministrar electricidad a todo el territorio nacional. La desventaja principal es que se necesita directa transimisión entre los distintos generadores y consumidores; desventaja que solamente se agrava considerando lo extenso que es Chile, incurriendo en crecientes pérdidas de energía entre más extensa sea la línea de transmisión[4]. La consecuencia directa de lo mencionado anteriormente es que varias localidades no reciben electricidad ya que los costos para implementar una línea de transmisión directa son prohibitivos.

Se necesita de otro paradigma de GTD para poder solucionar la falta de energía para estas localidades más distanciadas de las grandes urbes o centros industriales. Es por esto que toma más fuerza el concepto de GTD pero aplicado a nivel local para así no requerir de una línea de transmisión desde los generadores a las propias localidades.

## 2.2. Micro-red

Las micro-redes son un paradigma distinto a los métodos tradicionales de GTD, donde el sistema puede ser considerado local y en cierto grado autónomo. Una definición técnica de lo que es una micro-red es la siguiente<sup>5</sup>[5]: "*Un grupo de cargas y recursos energéticos distribuidos interconectados dentro de una zona eléctrica claramente delimitada, actuando como una entidad individualmente controlable con respecto a la red*<sup>6</sup>. Una micro-red puede conectarse o desconectarse de la red para permitir la operación tanto en modo conectado<sup>7</sup> como en modo isla<sup>8</sup>. Se da a entender que incluso con la ambigüedad de la "localidad" de la definición, la característica principal es la capacidad de desconectarse de la red principal de un estado y poder operar en modo isla.

Las micro-redes están recibiendo cada vez más atención a la hora de implementar nueva capacidad energética debido a que ofrecen ventajas[6] con respecto a métodos de GTD

---

<sup>3</sup>Cada generadora requiere de características geográficas preexistentes para su funcionamiento. Por ejemplo, recursos fluviales en el caso de hidroeléctricas, zonas costeras para el suministro de carbón a termoeléctricas, zonas de alta radiación solar en el caso de paneles solares, entre otros.

<sup>4</sup>Conocidos coloquialmente como apagones.

<sup>5</sup>La fuente está escrita en inglés por ende el autor de este documento presenta su propia traducción al español de la definición.

<sup>6</sup>En este caso entiéndase red como una red eléctrica de envergadura extensa.

<sup>7</sup>Aunque la traducción de *grid-connected* es literalmente *red-conectado*, el autor se da la libertad de simplificar la traducción y ocupar esta terminología de ahora en adelante

<sup>8</sup>island-mode se traduce literalmente como *modo-isla* pero nuevamente el autor se da la libertad de modificar la traducción para mejor lectura.

tradicionales:

- Resiliencia.
- Confiabilidad.
- Disminución de costos.
- Energía limpia<sup>9</sup>.

### 2.2.1. Componentes en una micro-red

Una micro-red ideal es capaz de poder tener todos sus consumidores y generadores eléctricos de manera interconectada y para lograr completa autonomía la potencia efectiva generada debiese ser mayor a la consumida en todo momento. Ya se mencionó con anterioridad que el concepto de micro-red es independiente del tipo de generador (o generadores) implementados, o sea, un generador termoeléctrico local es tan admisible como un conjunto de paneles solares. Habiendo dicho esto, se pueden encontrar ciertos componentes generales que toda micro-red debe incorporar para su correcto funcionamiento.

En el aspecto más básico de una micro-red, esta solo consta de 3 componentes fundamentales para su correcta operación:

- Generador.
- Consumidor.
- Almacenador

Todos los componentes mencionados anteriormente conectados a una red local de transmisión de energía. Otro componente que se considera en la literatura es la propia red de distribución nacional y la capacidad de conectarse a ella[7]. En este trabajo se concentrará en los primeros tres componentes mencionados con anterioridad, efectivamente considerando que la micro-red a emular está en modo isla. La figura 2.1 presenta un ejemplo de micro-red con los componentes más básicos posibles dentro de la misma.

Los generadores son todos aquellos componentes que aportan potencia activa a la micro-red haciendo uso de distintos métodos de generación eléctrica. La definición de micro-red es no restrictiva a la hora de elegir una tecnología por sobre otra, es decir, una micro-red cualquiera no prefiere paneles solares a un método de generación fósil, por ejemplo. Habiendo dicho esto, la tendencia a la hora de implementar generación para las micro-redes tiende a ser tecnología ERNC debido a un actual desarrollo e investigación de esta tecnología. Además estas tecnologías tienen la característica de que sus costos de instalación son reducidos<sup>10</sup> permitiendo su instalación en zonas remotas del planeta. Debido a que los componentes

---

<sup>9</sup>En caso de ocupar ERNC.

<sup>10</sup>Que tan costoso puede llegar a ser instalar un panel o un aerogenerador es relativo pero ciertamente menos costoso que instalar una instalación hidroeléctrica o termoeléctrica completa.

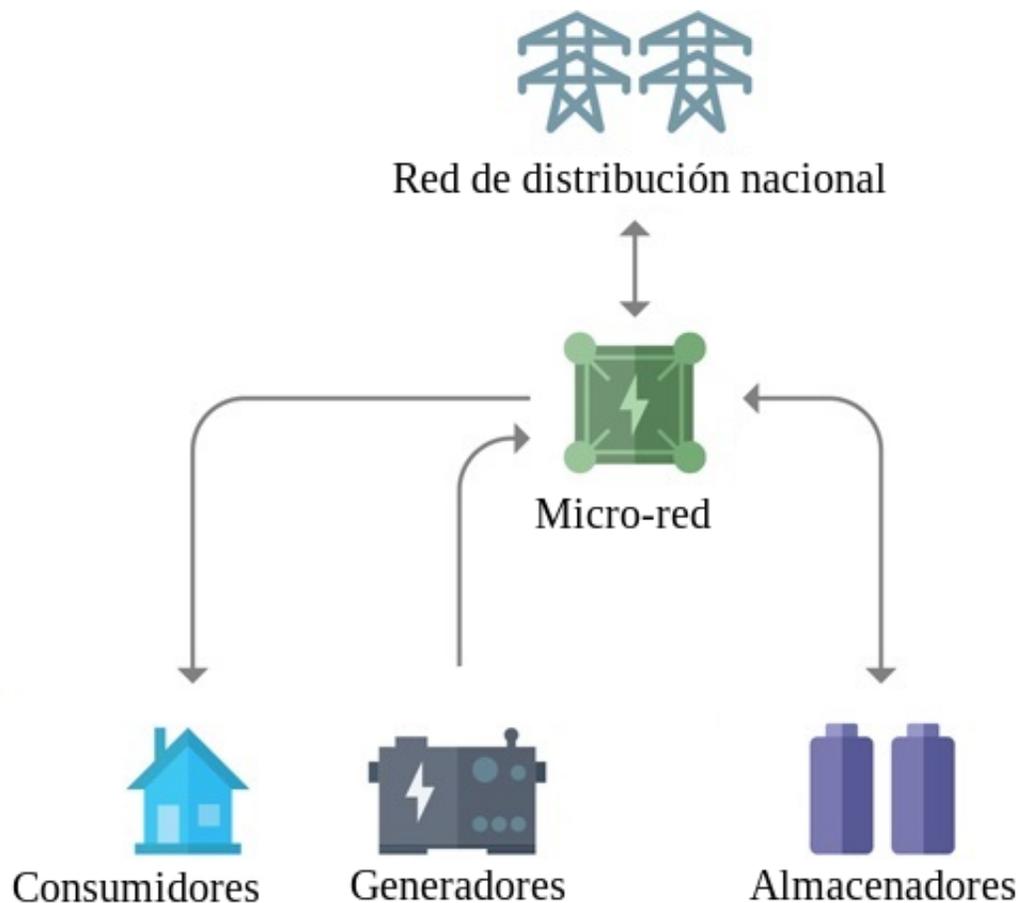


Figura 2.1: Micro-red básica. Notar que la red de distribución nacional fue incorporada solo por completitud y no será considerada un componente existente para el desarrollo de la micro-red a emular. Imagen editada de [1]

ERNC no permiten que la generación de potencia activa sea constante a través del tiempo sino que fluctúa de forma extrema dependiendo de distintos factores atmosféricos, es que se tiende a agregar un generador adicional en base a tecnología fósil<sup>11</sup> el cual es capaz de suplir la falta de potencia que requieren los consumidores.

Dentro de la categoría de consumidores puede existir un sinnúmero de posibilidades: zonas residenciales, comerciales o industriales; edificaciones y servicios públicos; con cada categoría pudiendo ser granulada aún más. Efectivamente en esta categoría se considera a todos los elementos de la red que sólo consumen potencia activa. Vale mencionar que actualmente la mayoría de micro-redes instaladas sirven el propósito de alimentar zonas residenciales de poca densidad pero esfuerzos se han hecho para poder crear micro-redes con la capacidad de mantener zonas industriales y todos los proceso productivos que ellas implican[8].

<sup>11</sup>Generadores diesel siendo bastante populares para este propósito.

## 2.3. *Data Distribution Service*

*Data Distribution Service* (DDS) es un *middleware*<sup>12</sup> de comunicación enfocado a mensajería máquina a máquina, creado por *Object Management Group*[9]. Esta tecnología se basa principalmente en la mensajería en base a suscriptores y publicadores. Una unidad de procesamiento<sup>13</sup> la cual tenga el protocolo instalado en su sistema tiene la capacidad de suscribirse o publicar a tópicos, siendo estos una abstracción del lugar en el que se manejaran los datos. De esta forma, existiendo distintos tópicos para distintos datos de interés, los nodos pueden elegir si suscribirse o publicar (o ambos) a cualquier tópico.

DDS no sólo se trata de garantizar el correcto intercambio de información entre nodos sino que también garantizando cierta calidad de servicio o *Quality of Service* (QoS) que exige que los elementos involucrados para una arquitectura DDS deban comportarse bajo parámetros de comunicación deseados, ofreciendo flexibilidad de como los componentes interactúan unos con otros.

Los elementos básicos existentes que conforman la DDS son los siguientes:

- Tópico: Representa información sobre un solo tipo de dato<sup>14</sup>, su distribución hacia la red y disponibilidad de muestras todavía no procesadas o requeridas.
- Editor de datos<sup>15</sup>: Genera muestras de un tipo de dato en particular.
- Lector de datos<sup>16</sup>: Recibe muestras de un tipo de dato en particular.
- Publicador: Aplica control y restricciones al manejo de datos a los que incurren sus editores de datos asociados.
- Suscriptor: Aplica control y restricciones al manejo de datos a los que incurren sus lectores de datos asociados.

Estos son los elementos que definen la transmisión de datos dentro de una red basada en DDS donde cada elemento puede tener parámetros QoS distintos a los demás, permitiendo flexibilidad a la hora de implementar una red de este tipo. Para una mejor noción de como los distintos elementos de esta red interactúan entre ellos es que se presentan las siguientes restricciones:

- Un publicador puede estar asociado a muchos editores de datos.
- Un suscriptor puede estar asociado a muchos lectores de datos.

---

<sup>12</sup>Middleware se define como software que se encuentra entre el sistema operativo y la capa de aplicación

<sup>13</sup>Nos referiremos a esta entidad como nodo.

<sup>14</sup>Tipo de dato en este contexto significa la característica principal con la que designan ciertos tipos de datos con respecto a otros. Por ejemplo, ciertos mensajes pueden representar los datos de potencia activa de una red mientras que otros representan la potencia reactiva de la misma. Lo anterior implica que los datos pertenecen a tópicos distintos debido a que el tipo de dato que representan es distinto.

<sup>15</sup>El autor de este trabajo se da la libertad de ocupar su propia traducción de *Data Writer*

<sup>16</sup>Nuevamente el autor se da la libertad de ocupar su propia traducción de *Data Reader*

- Editores de datos pueden estar asociados a un solo tópico.
- Lectores de datos pueden estar asociados a un solo tópico.
- Un tópico puede estar asociado a muchos editores y lectores de datos.

A modo de ejemplo, una manera común de envío y recepción de información es que el publicador está directamente conectado con una aplicación<sup>17</sup>, la cual le envía los datos relevantes los cuales son traspasados a sus editores de datos. Los editores de datos reciben estas muestras y las envían al tópico relevante (el cual este asociado al tipo de dato de las muestras). Aquí el tópico puede mandar una señal de que han llegado nuevas muestras a los lectores de datos con el tópico o también los lectores de datos pueden verificar periódicamente si es que hay nuevos datos en su tópico, ambos modos de operar son válidos. Con los datos adquiridos el lector de dato le traspasa las muestras al suscriptor el cual luego procesa los datos y los envía a otra aplicación no necesariamente igual a la antes mencionada. La figura 2.2 muestra como los elementos que la integran interactúan entre sí.

Con el ejemplo antes mencionado quizás no se puede apreciar el traspaso de información que ocurrió debido a que se puede percibir que el traspaso de información ocurrió de manera local, en un mismo dispositivo pero vale recordar que los tópicos pueden estar asociados a muchos editores y lectores de datos, estos pueden estar en distintos dispositivos distribuidos por la red. Además el ejemplo muestra como se puede desacoplar todo lo que es DDS con la aplicación en sí misma, esto ofrece una ventaja de modularidad donde cambios hechos en la morfología de la red no implica un cambio en como debe operar la aplicación de interés.

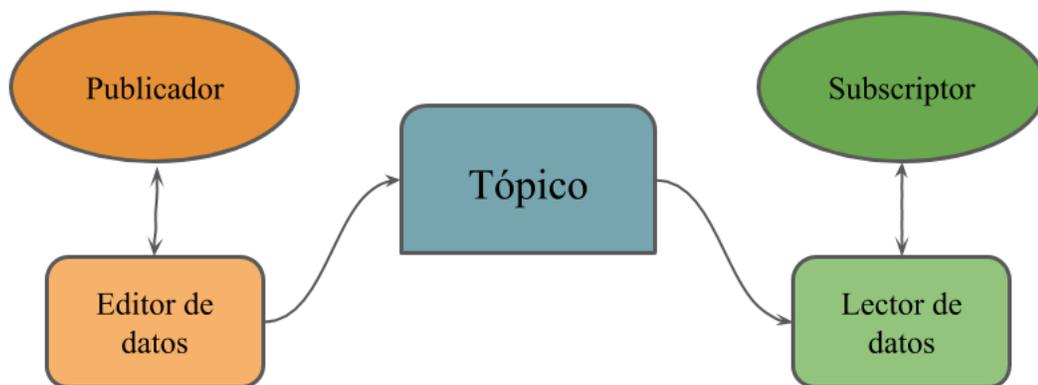


Figura 2.2: Arquitectura básica de cómo opera una red de información en base a DDS.

<sup>17</sup>Se refiere a aplicación como una operación de interés, como por ejemplo control de actuadores, reporte de muestras, etcétera.

### 2.3.1. Real-Time Publish-Subscribe

Dentro de DDS existe el protocolo Real-Time Publish-Subscribe (RTPS) el cual sus principales objetivos son [10]:

- Tolerancia de fallos ante pérdida de paquetes.
- Extensibilidad para hacer al propio protocolo compatible con futuras versiones de si mismo.
- Permitir modularidad para lograr que nuevas aplicaciones puedan ser integradas sin necesidad de cambiar la arquitectura de fondo.
- Acceso a configuraciones hacia la red misma para comprobar los requerimientos de seguridad y confiabilidad de la red.
- Escalabilidad de la red de información.

Este protocolo embebido en DDS también cuenta con módulos propios que hacen las tareas antes descritas. No es de importancia para el trabajo propuesto en este documento analizar estos módulos exceptuando solo uno: El módulo *Discovery*

*Discovery* será el modulo a ocupar en el proyecto para lograr descubrimiento de nodos descentralizados, evitando así tener que ocupar un dispositivo central que se encargue de este proceso en particular. La forma en que este módulo puede descubrir de forma no centralizada depende netamente de la implementación DDS a ocupar, ya se verá que se hará uso OpenDDS, una implementación de código abierto de DDS, el cual hace uso de los lectores y editores de datos de la red para hacer reconocimiento de los distintos elementos de la red[11].

### 2.3.2. Implementaciones de DDS

DDS no es nada más que es un estándar, una lista de definiciones y descripciones para generar un sistema de comunicación. Es entonces que existen distintas implementaciones de este protocolo, en particular, OpenDDS, su logo presente en la figura 2.3.

Dentro de las implementaciones existentes destacan tres:

1. *Connex DDS*[12].
2. *Vortex OpenSplice*[13].
3. *OpenDDS*[14].

El autor presenta estas alternativas posibles para la implementación de un sistema de intercambio de información debido a que no se generó un análisis exhaustivo<sup>18</sup> de cual ocupar

---

<sup>18</sup>Exhaustivo en el sentido de probar las tres implementaciones y analizar como se comportan en una misma prueba o marco de referencia. Lo anterior, tiene sentido solamente si es que este trabajo se basa en ver los

y se limitó a probar las alternativas presentadas. La elección de la alternativa a ocupar se basó meramente en:

1. Estado de desarrollo.<sup>19</sup>
2. Facilidad de instalación del *software*.
3. Disponibilidad del código fuente.

Una rápida inspección da a cuenta que las tres implementaciones siguen siendo desarrolladas y mejorándose con cada versión. Este trabajo no requiere todas las funciones que DDS describe, solo se limita a requerir la capacidad de envío y recepción de datos de interés de forma ordenada. Es por esto que el estado de desarrollo de estas tres alternativas es suficiente para efectos de lo que se quiere implementar.

Debido a como se verá en 2.4.1, el computador está hecho de una arquitectura que difiere de arquitecturas más másificadas como la existentes en la gran mayoría de computadores domésticos. Es por lo anterior que cabe la duda si es que posible equipar los computadores de interés con el *software* necesario. De las tres implementaciones, fue solo *Connex DDS* la que no se pudo instalar. El autor ignora la razón del porqué de esto.

Para la investigación es de especial importancia poder acceder al código fuente de cualquier *software* que utiliza si es que nace la necesidad de seguir desarrollando o innovando, permitiendo la capacidad de modificar dicho *software* como se le plazca. De las restantes alternativas, ambas permiten el acceso a su código fuente pero *Vortex OpenSplice* presenta tener dos versiones de su producto: uno de comunidad y otro profesional. El código fuente que permite esta implementación es el de la versión comunidad, lo que implica que no revela todas las funcionalidades posibles, limitando futuros desarrollos sobre esta plataforma.

Teniendo en cuenta todo lo mencionado anteriormente es que el autor prefiere hacer uso de *OpenDDS* a la hora de desarrollar el trabajo a presentar. Esta plataforma sigue siendo actualizada, fue posible ser instalada y permite el acceso y edición de su código fuente[15].



Figura 2.3: Logo de *OpenDDS*.

---

efectos de distintas tecnologías DDS sobre un problema de implementación específico, lo cual no se trata este trabajo.

<sup>19</sup>Esto significa dos cosas, si es que la implementación en sí ha llegado a un punto donde es factible ocuparla para propósito de este trabajo y también si es que el equipo desarrollador sigue activo.

## 2.4. *Hardware*

Debido a que se necesita de implementar una red de comunicación donde exista la transmisión de datos, es que se necesitan ocupar distintos dispositivos con distintas funciones cada uno.

### 2.4.1. Raspberry Pi

La Raspberry Pi , ilustrada en la figura 2.4 es una computadora reducida en tamaño, costo y consumo eléctrico que ofrece versatilidad a la hora de generar de diseñar un sistema distribuido e interconectado. Todavía está en desarrollo por lo cual existen muchas versiones de Raspberry Pi existentes, siendo el modelo 3B una de las últimas versiones existentes al momento de escribir este documento y la que se ocupará para propósitos de este trabajo.



Figura 2.4: Raspberry Pi Modelo 3B.

El requerimiento más importante que se necesita de este computador es la capacidad de poder instalar el protocolo DDS en su sistema, lo cual no es necesariamente posible considerando que la arquitectura del procesador de este dispositivo es ARM, contrastando la arquitectura de computadores domiciliarios que puede ser x64 o x86, por ende, es necesario tener esto en consideración a la hora de instalar el *software* necesario. Referase a 4.0.2 para las especificaciones de instalación. La raspberry Pi también incluye la capacidad de conexión *WiFi* incorporada, facilitando el desarrollo e implementación del sistema en su conjunto.

# Capítulo 3

## Metodología

En esta sección se presenta la forma en que se hará el proyecto, las consideraciones que se tomaron y propia implementación de la red emulada. Esta sección no detallará como se hacen procesos más bien técnicos de descarga, compilación e instalación del *software* requerido, sino que se enfocará principalmente en los datos a ocupar y en como se emulará la red en sí. Si el lector está más interesado en cómo comenzar a ocupar las distintas plataformas, lo recomendable es ir a la sección 4.0.2.

### 3.1. Datos para la emulación

El centro de energía de la universidad de Chile aportó con datos de potencia activa del periodo de septiembre de 2018 en la localidad de Huatacondo, donde existe implementada una micro-red real. Los datos incluyen cuatro categorías de importancia:

- PV.
- Demanda.
- Baterías.
- Genset.

Cada categoría presenta una tasa de muestreo de 10 segundos y 43201 muestras en total. Con estos datos se hará un breve análisis de su comportamiento y del como pueden ser ocupados para demostrar el correcto funcionamiento de la red a emular.

#### 3.1.1. PV

El principal método de generación eléctrica de Huatacondo son sus paneles fotovoltaicos que entregan aproximadamente 17 kW de generación en su pico, además permitiendo un com-

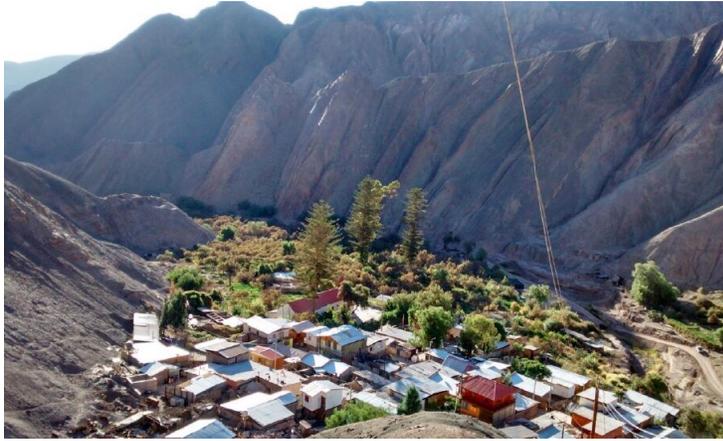


Figura 3.1: La localidad de Huatacondo.

portamiento de generación muy regular y periódico a través del mes completo, ejemplificado en la figura 3.2, principalmente debido a estar Huatacondo en una región del país con poca nubosidad y abundante radiación solar.

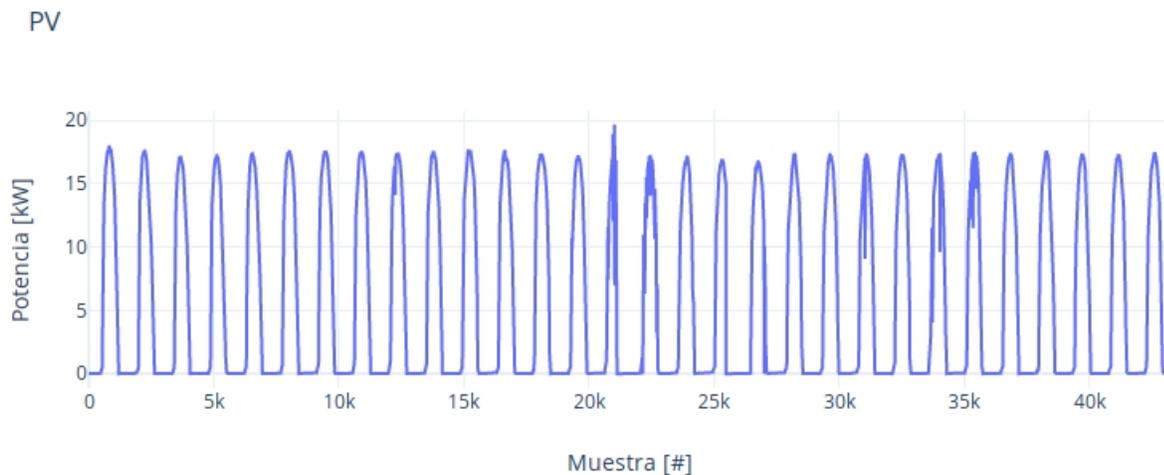


Figura 3.2: Potencia de generación fotovoltaica de la red.

### 3.1.2. Demanda

Esta categoría corresponde a la demanda completa de la red y siendo Huatacondo una localidad rural, se asume que corresponde a una demanda más bien residencial. El comportamiento de esta categoría es periódico donde se puede dilucidar que el periodo de máximo consumo energético de pasado el mediodía y con una tendencia a que el máximo de consumo por la época de festividades<sup>1</sup>. Para una mayor claridad de la potencia generada y consumida por la red se presentan ambos datos, donde es clara la necesidad de poder suplir la demanda con más generación.

<sup>1</sup>Aproximadamente por el 18 de septiembre.

### Demanda

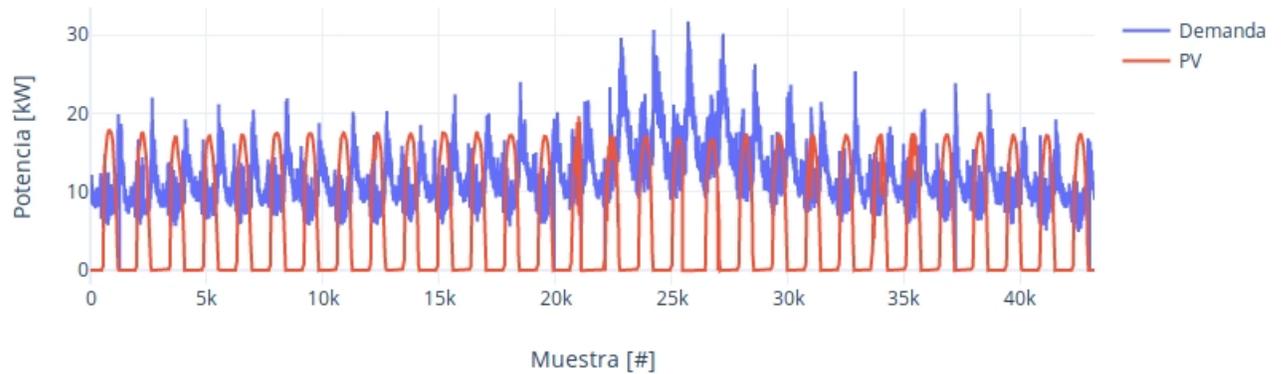


Figura 3.3: Perfil de la demanda de la red, superpuesta con la potencia entregada por los paneles fotovoltaicos.

### 3.1.3. Baterías

Las baterías cuando se recargan, su valor es negativo y cuando entregan potencia a la red, positivo. Su comportamiento es ciertamente periódico, como se aprecia en la figura 3.4, con claros cambios relativamente bruscos debido principalmente a la etapa de prendido y apagado del generador diesel existente en la red.

### Baterías



Figura 3.4: Potencia entregada y consumida de las baterías en la red.

### 3.1.4. Genset

El generador diesel presenta ser una alternativa eficaz de generación cuando los paneles solares no se encuentran activos durante la noche y las baterías o entregan muy poca potencia o se encuentran ellas mismas consumiendo energía. La capacidad de entregar potencia del

generador diesel supera con creces la capacidad de generación de los paneles fotovoltaicos tanto así que es posible proveer tanto a la demanda como cargar las baterías existentes. En la figura 3.5 se muestra el contraste de generación entre los paneles fotovoltaicos, la cual tiene un caracter diurno y el generador diesel, donde es requerido en los periodos nocturnos.



Figura 3.5: Potencia entregada por el generador diesel superpuesta con la potencia generada de los paneles fotovoltaicos.

### 3.1.5. Conclusión de los datos analizados

Los generadores presentan ser parte vital en el correcto de la red, permiten que la demanda de potencia sea entregada y que las baterías puedan ser cargadas. Se puede observar que las batería entregan el máximo de potencia a la red cuando en los tiempos en que los paneles solares dejan de producir energía suficiente y cuando los mismos paneles comienzan a producir energía de nuevo. La figura 3.6 muestra como se comportan todos los elementos de la red al mismo tiempo en tema de potencia donde se observa que existe importancia en el manejo del generador diesel debido a toda la potencia que suministra a la red.

Debido a la importancia de como opera el generador diesel, es que el autor propone ocupar estos datos en particular, como prueba de concepto, o sea, ocupar los datos de los paneles fotovoltaicos, la demanda y las baterías e incorporarlos a una red de información basada en DDS. Con estos datos se puede asumir cuanta energía puede requerir la red para su correcto funcionamiento y cuando debiese comenzar a operar de forma automática el generador eléctrico, teniendo los datos originales de generación de potencia en base a diesel como base para corroborar los datos.

## 3.2. Arquitectura de la red

Debido a que existen cuatro elementos en la red pero se ocuparán sólo tres para el envío de datos, se hará uso de tres Raspberry Pi modelo 3B con 16GB de tarjeta microSD. La tarea

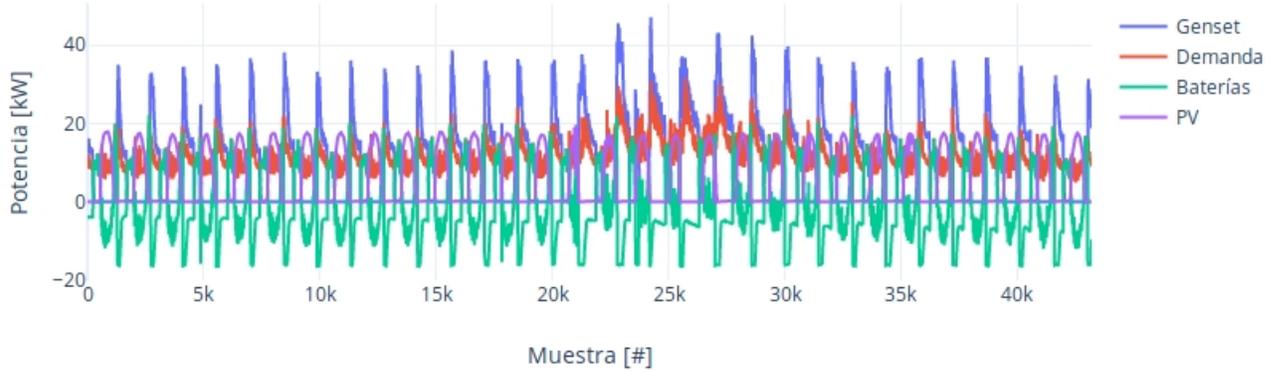


Figura 3.6: Potencia combinada de todos los elementos de la red.

de estos dispositivos será la de emular a: Demanda, PV y a Baterías. Los datos de potencia se encapsularán en una variable tipo *float*, los cuales serán de forma periódica enviados por el publicador al tópico de datos de potencia. Se puede concluir que debido a que sólo se maneja un tipo de dato en la red (el de potencia encapsulado en la variable *float*) es que es necesario ocupar sólo un tópico para toda la red, con todos los editores y lectores de datos haciendo uso extensivo de este elemento.

En esta red se necesitan entonces tres publicadores, los cuales se encargarán de regular el envío de información hacia el tópico, y un subscriptor, donde su objetivo es extraer los datos del tópico y hacer un análisis de cuanta energía necesita el sistema y si es por ende necesario iniciar el generador diesel. La fórmula para hacer saber cuanta energía requiere el sistema y por ende, cuanta energía debe entregar el generador diesel es la siguiente:

$$P_{generador} = P_{demanda} - P_{PV} - P_{baterias} \quad (3.1)$$

En caso de que los datos recibidos por el subscriptor impliquen que  $P_{generador} < 0$ , entonces se debe generar un aviso de prendido de generador diesel y estipular cuanta corriente se necesita. Con el dato estimado de potencia requerida se puede hacer una comparación con los datos reales de potencia requerida, esta comparación permitirá ver si es que el sistema funciona correctamente o con algún grado de error.

En 3.7 se obvía los editores y lectores de datos debido a que se puede asumir su existencia debido a que:

- Cada publicador tiene su único editor de datos asociado.
- Cada subscriptor tiene su único lector de datos asociado.
- Dentro de cada dispositivo pueden coexistir un subscriptor y publicador en forma paralela (con su editor y lector de datos asociados).

Los tres dispositivos entonces cumplen roles distintos dentro de la red:

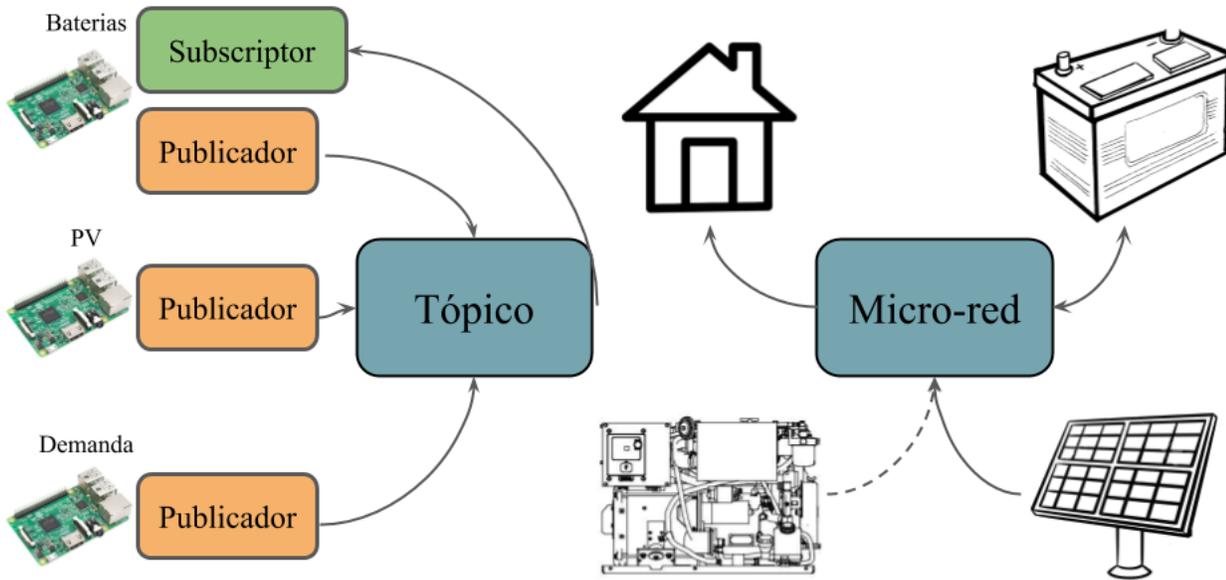


Figura 3.7: Comparación entre la micro-red real y la emulación propuesta.

- El dispositivo asociado a Demanda solo envía al tópico los datos de Demanda.
- El dispositivo asociado a PV solo envía al tópico los datos de PV
- Por último el dispositivo asociado a Baterías sirve un doble propósito, enviar los datos de Baterías pero también recibir las muestras existentes en el tópico y discernir si es que es necesario prender el generador diesel.

Se puede dilucidar entonces que los datos son enviados por en intervalos de diez segundos por los publicadores (estos recibiendo las muestras mediante una aplicación), el tópico entonces empieza a llenarse con muestras todavía no analizadas y es aquí cuando el subscriber pide los datos al tópico. Para hacer las cosas más sencillas, la ecuación 3.1 se verá implementada en el lector de datos asociado al subscriber, o sea, no será necesario implementar hacer otra aplicación donde exista la capacidad de discernir si es que se necesita el generador prendido y con cuanta potencia.

Vale decir que tanto el dispositivo que emula los datos de PV y Demanda también tienen un subscriber coexistiendo con su publicador solo que este se encuentra desactivado y no aporta nada a la red. Lo anterior es debido a que aunque los nodos (en especial, Baterías) sirven propósitos distintos, el código base para generar sus elementos DDS es el mismo, haciendo uso de una librería común basada en OpenDDS creada por el autor de este documento, denominada *libCommonThesis.so*. La librería antes mencionada promete poder habilitar dispositivos para intercambio de información tipo publicador-subscriber de forma clara y sencilla.

Las tres Raspberry Pi están conectadas mediante *WiFi* a una *Wireless Local Area Network*

(WLAN) donde la probabilidad de pérdida de paquetes es reducida. Si este proyecto es considerado para futuros trabajos es importante generar conexiones remotas.

Por último, para poder discernir que dispositivo envía dato al tópico, es que a cada dispositivo se le asocia un ID con el cual el lector de datos puede hacer la distinción sobre de cual dispositivo proviene la muestra analizada.

### 3.2.1. Descubrimiento descentralizado

Para habilitar el descubrimiento no centralizado de elementos DDS, solo basta con cambiar las condiciones de conexión con las que los nodos son generados. OpenDDS ofrece una manera sencilla de cambiar estos parámetros, basta con hacer un archivo donde se especifique el protocolo de transporte a ocupar y el método de descubrimiento. El archivo generado se denomina *rtps.ini* y contiene lo siguiente:

```
[common]
DCPSGlobalTransportConfig=$file
DCPSDefaultDiscovery=DEFAULT_RTPS
```

```
[transport/the_rtps_transport]
transport_type=rtps_tcp
```

# Capítulo 4

## Resultados

En este capítulo se muestran los distintos resultados obtenidos con las pruebas expuestas en 3. Las pruebas consideran lo que entrega el lector de datos del dispositivo asociado con Baterías. La forma en la que el lector de datos entrega la información y obteniendo los datos de cada dispositivo, ordenarlos y luego desplegarlos en el terminal.

Para estas pruebas se quisieron emular dos caso distintos: Cuando es necesario que el generador diesel esté prendido y cuando no. Llegar a estos datos usando la tasa de envío de diez segundos hubiese sido engorroso, así que se decidió ocupar un atajo a la hora de probar la red, ignorando las muestras anteriores y comenzando inmediatamente en la zona de interés. Técnicamente, hacer una prueba completamente real con los datos entregados tomaría aproximadamente un mes.

Como se ve en 4.1, las muestras Demanda, PV y Baterías fueron recibidas satisfactoriamente. Genset ya existía en los datos así que se extrajo directamente para comparar con `Genset_predict`, el cual es la predicción hecha por la ecuación 3.1. Se puede observar que la predicción es correcta pero debido mayormente a que los datos recibidos para llegar a ella, llegaron de forma correcta.

De forma similar, se quizó probar en una zona donde ya se supiese que el generador no estuviese activo. Los resultados se pueden verificar en 4.2 donde Genset está completamente desactivado pero su predicción fluctúa en valores menores. El autor considera que esa variación es debido principalmente a las limitaciones de ocupar una variable *float* para manejar datos, es posible que el uso de una variable tipo *double* corrija este fenómeno. Aun así, se puede apreciar que la predicción está muy cerca a cero, el valor real de la potencia entregada por el generador.

### 4.0.1. Análisis

Para ambos casos propuestos, se logró que los datos se transfiriesen de manera correcta, generando estimaciones correctos de como debería comportarse el generador diesel. Esto comprueba que la arquitectura propuesta para emular la micro-red presenta ser efectiva a la

File	Edit	View	Search	Terminal	Help
Demanda=11.7029	PV=0.0278856	Genset=15.6312	Baterias=-3.95618	Genset_predict=15.6312	
Demanda=12.2261	PV=0.0318947	Genset=16.131	Baterias=-3.93682	Genset_predict=16.131	
Demanda=12.0818	PV=0.0277416	Genset=15.9985	Baterias=-3.94442	Genset_predict=15.9985	
Demanda=11.9395	PV=0.0320101	Genset=15.8762	Baterias=-3.96871	Genset_predict=15.8762	
Demanda=12.0213	PV=0.0271842	Genset=15.9545	Baterias=-3.96038	Genset_predict=15.9545	
Demanda=12.0748	PV=0.0316928	Genset=15.9958	Baterias=-3.95267	Genset_predict=15.9958	
Demanda=10.3705	PV=0.028194	Genset=14.2993	Baterias=-3.95705	Genset_predict=14.2993	
Demanda=10.0488	PV=0.0313462	Genset=13.937	Baterias=-3.91953	Genset_predict=13.937	
Demanda=9.55459	PV=0.0284458	Genset=13.4177	Baterias=-3.89152	Genset_predict=13.4177	
Demanda=9.77204	PV=0.0303406	Genset=13.6725	Baterias=-3.9308	Genset_predict=13.6725	
Demanda=9.68872	PV=0.0318964	Genset=13.5892	Baterias=-3.93243	Genset_predict=13.5892	
Demanda=9.89235	PV=0.0294981	Genset=13.8096	Baterias=-3.94675	Genset_predict=13.8096	
Demanda=9.77983	PV=0.0298735	Genset=13.6625	Baterias=-3.91254	Genset_predict=13.6625	
Demanda=10.3003	PV=0.0318689	Genset=14.2112	Baterias=-3.94276	Genset_predict=14.2112	
Demanda=10.2344	PV=0.0300079	Genset=14.1388	Baterias=-3.93445	Genset_predict=14.1388	
Demanda=10.1033	PV=0.0315484	Genset=14.0055	Baterias=-3.93378	Genset_predict=14.0055	
Demanda=10.0034	PV=0.0308412	Genset=13.9272	Baterias=-3.95467	Genset_predict=13.9272	
Demanda=9.86248	PV=0.0302706	Genset=13.7506	Baterias=-3.91839	Genset_predict=13.7506	
Demanda=9.1109	PV=0.0292097	Genset=14.0055	Baterias=-3.92376	Genset_predict=14.0055	
Demanda=9.97361	PV=0.033455	Genset=13.8802	Baterias=-3.94005	Genset_predict=13.8802	
Demanda=9.62634	PV=0.032083	Genset=13.5276	Baterias=-3.93335	Genset_predict=13.5276	
Demanda=8.90821	PV=0.0295968	Genset=12.8338	Baterias=-3.95519	Genset_predict=12.8338	
Demanda=8.62883	PV=0.0325766	Genset=12.5362	Baterias=-3.93991	Genset_predict=12.5362	
Demanda=8.49755	PV=0.0318878	Genset=12.399	Baterias=-3.93334	Genset_predict=12.399	
Demanda=8.45454	PV=0.0254363	Genset=12.3402	Baterias=-3.91106	Genset_predict=12.3402	
Demanda=8.84636	PV=0.0278537	Genset=12.7078	Baterias=-3.88925	Genset_predict=12.7078	
Demanda=9.1513	PV=0.030942	Genset=13.0308	Baterias=-3.91039	Genset_predict=13.0307	
Demanda=9.09035	PV=0.0300207	Genset=12.9718	Baterias=-3.91142	Genset_predict=12.9718	
Demanda=8.93598	PV=0.030051	Genset=12.84	Baterias=-3.93407	Genset_predict=12.84	
Demanda=8.79379	PV=0.0319539	Genset=12.7123	Baterias=-3.9505	Genset_predict=12.7123	
Demanda=8.98689	PV=0.0275385	Genset=12.8692	Baterias=-3.90985	Genset_predict=12.8692	
Demanda=9.3366	PV=0.0333	Genset=13.2454	Baterias=-3.9421	Genset_predict=13.2454	
Demanda=9.50116	PV=0.0310609	Genset=13.408	Baterias=-3.93791	Genset_predict=13.408	
Demanda=9.71754	PV=0.0287958	Genset=13.648	Baterias=-3.95925	Genset_predict=13.648	
Demanda=9.59421	PV=0.0312546	Genset=13.4765	Baterias=-3.91354	Genset_predict=13.4765	
Demanda=9.38233	PV=0.0271411	Genset=13.2317	Baterias=-3.87648	Genset_predict=13.2317	
Demanda=9.34257	PV=0.0307316	Genset=13.261	Baterias=-3.94916	Genset_predict=13.261	
Demanda=9.48237	PV=0.0305733	Genset=13.354	Baterias=-3.9022	Genset_predict=13.354	
Demanda=9.84426	PV=0.0296892	Genset=13.739	Baterias=-3.92443	Genset_predict=13.739	
Demanda=9.48072	PV=0.0315924	Genset=13.3835	Baterias=-3.93437	Genset_predict=13.3835	
Demanda=9.68548	PV=0.0296433	Genset=13.5647	Baterias=-3.90883	Genset_predict=13.5647	
Demanda=9.80938	PV=0.0292205	Genset=13.7655	Baterias=-3.98534	Genset_predict=13.7655	
Demanda=9.6769	PV=0.027646	Genset=13.5843	Baterias=-3.93508	Genset_predict=13.5843	
Demanda=9.27378	PV=0.03115	Genset=13.212	Baterias=-3.96937	Genset_predict=13.212	
Demanda=9.31444	PV=0.0269589	Genset=13.2454	Baterias=-3.95792	Genset_predict=13.2454	
Demanda=9.07132	PV=0.0282601	Genset=12.9748	Baterias=-3.93174	Genset_predict=12.9748	

Figura 4.1: Datos recibidos y desplegados por el lector de datos. Caso cuando el generador si se encuentra activo en la red. Los datos representan valores en [kW]

hora de recibir y enviar datos, también se comprueba la capacidad de DDS para generar un sistema de transmisión extremadamente modular.

#### 4.0.2. Errores aleatorios de conexión

Aunque los resultados obtenidos anteriormente muestran como la red puede enviar y entregar datos de forma segura, en algunas ocasiones ocurría una pérdida de conexión de la cual por lo menos el subscriptor no podía recuperarse. El autor del documento tiene sospechas de que puede estar causando tales problemas pero hasta ahora no tiene una razón concreta de que puede estar pasando en estos casos.

File	Edit	View	Search	Terminal	Help
Demanda=7.24019	PV=16.6107	Genset=0	Baterias=-9.37049	Genset_predict=-9.53674e-07	
Demanda=7.67819	PV=16.4874	Genset=0	Baterias=-8.80921	Genset_predict=0	
Demanda=7.85202	PV=16.4747	Genset=0	Baterias=-8.6227	Genset_predict=9.53674e-07	
Demanda=7.73044	PV=16.5377	Genset=0	Baterias=-8.80726	Genset_predict=0	
Demanda=8.11657	PV=16.4844	Genset=0	Baterias=-8.36782	Genset_predict=-9.53674e-07	
Demanda=8.66176	PV=16.5158	Genset=0	Baterias=-7.85408	Genset_predict=9.53674e-07	
Demanda=8.37782	PV=16.4981	Genset=0	Baterias=-8.12023	Genset_predict=0	
Demanda=8.23184	PV=16.4709	Genset=0	Baterias=-8.23901	Genset_predict=-9.53674e-07	
Demanda=8.82186	PV=16.4965	Genset=0	Baterias=-7.67467	Genset_predict=-4.76837e-07	
Demanda=9.6499	PV=16.4798	Genset=0	Baterias=-6.82994	Genset_predict=0	
Demanda=10.4024	PV=16.5145	Genset=0	Baterias=-6.11216	Genset_predict=-9.53674e-07	
Demanda=10.2959	PV=16.4694	Genset=0	Baterias=-6.17348	Genset_predict=-9.53674e-07	
Demanda=10.2073	PV=16.4375	Genset=0	Baterias=-6.23027	Genset_predict=0	
Demanda=9.51975	PV=16.4079	Genset=0	Baterias=-6.88811	Genset_predict=0	
Demanda=9.20534	PV=16.3365	Genset=0	Baterias=-7.1312	Genset_predict=-4.76837e-07	
Demanda=9.26534	PV=16.3237	Genset=0	Baterias=-7.05839	Genset_predict=-4.76837e-07	
Demanda=8.81727	PV=16.2683	Genset=0	Baterias=-7.45103	Genset_predict=0	
Demanda=8.62321	PV=16.2804	Genset=0	Baterias=-7.65722	Genset_predict=9.53674e-07	
Demanda=8.60549	PV=16.2726	Genset=0	Baterias=-7.66716	Genset_predict=9.53674e-07	
Demanda=8.59951	PV=16.1637	Genset=0	Baterias=-7.5642	Genset_predict=4.76837e-07	
Demanda=8.21651	PV=16.1092	Genset=0	Baterias=-7.89273	Genset_predict=-9.53674e-07	
Demanda=8.64707	PV=16.0698	Genset=0	Baterias=-7.42277	Genset_predict=-4.76837e-07	
Demanda=9.11025	PV=16.0903	Genset=0	Baterias=-6.98006	Genset_predict=4.76837e-07	
Demanda=10.4365	PV=16.1273	Genset=0	Baterias=-5.69075	Genset_predict=-9.53674e-07	
Demanda=10.4873	PV=16.1284	Genset=0	Baterias=-5.64117	Genset_predict=4.76837e-07	
Demanda=9.79428	PV=16.0954	Genset=0	Baterias=-6.30113	Genset_predict=4.76837e-07	
Demanda=8.52202	PV=16.0749	Genset=0	Baterias=-7.55284	Genset_predict=-4.76837e-07	
Demanda=8.33062	PV=16.0568	Genset=0	Baterias=-7.72615	Genset_predict=-9.53674e-07	
Demanda=8.01388	PV=16.0217	Genset=0	Baterias=-8.00777	Genset_predict=-9.53674e-07	
Demanda=7.92934	PV=15.96	Genset=0	Baterias=-8.03069	Genset_predict=9.53674e-07	
Demanda=8.13465	PV=15.9704	Genset=0	Baterias=-7.83573	Genset_predict=0	
Demanda=7.70052	PV=15.9682	Genset=0	Baterias=-8.2677	Genset_predict=0	
Demanda=7.54738	PV=15.9245	Genset=0	Baterias=-8.37715	Genset_predict=0	
Demanda=7.93123	PV=15.8288	Genset=0	Baterias=-7.89756	Genset_predict=0	
Demanda=8.29558	PV=15.8383	Genset=0	Baterias=-7.5427	Genset_predict=4.76837e-07	
Demanda=7.94502	PV=15.8147	Genset=0	Baterias=-7.86967	Genset_predict=0	
Demanda=7.95297	PV=15.8079	Genset=0	Baterias=-7.85493	Genset_predict=0	
Demanda=8.00354	PV=15.7858	Genset=0	Baterias=-7.78223	Genset_predict=-4.76837e-07	
Demanda=8.25429	PV=15.7643	Genset=0	Baterias=-7.51002	Genset_predict=0	
Demanda=8.4657	PV=15.6871	Genset=0	Baterias=-7.22138	Genset_predict=-4.76837e-07	
Demanda=8.33463	PV=15.6692	Genset=0	Baterias=-7.33453	Genset_predict=-4.76837e-07	
Demanda=8.5297	PV=15.6775	Genset=0	Baterias=-7.14782	Genset_predict=0	
Demanda=8.37786	PV=15.571	Genset=0	Baterias=-7.19315	Genset_predict=0	
Demanda=8.1515	PV=15.545	Genset=0	Baterias=-7.39346	Genset_predict=-4.76837e-07	
Demanda=8.12312	PV=15.4989	Genset=0	Baterias=-7.37574	Genset_predict=-4.76837e-07	
Demanda=7.67711	PV=15.4073	Genset=0	Baterias=-7.73022	Genset_predict=0	
Demanda=8.23888	PV=15.3734	Genset=0	Baterias=-7.13455	Genset_predict=4.76837e-07	

Figura 4.2: Datos recibidos y desplegados por el lector de datos. Caso cuando el generador si se encuentra activo en la red. Los datos representan valores en [kW]

# Conclusión

Como se puede observar en el presente documento, se logra crear una red de intercambio de información para pruebas de concepto, los datos de Huatacondo son transmitidos y compartidos entre los distintos elementos de la red emulada, permitiendo corroborar que la estimación implementada logra resultados positivos. El sistema publicador y subscriber que otorga DDS logra entonces ser un candidato capaz y eficiente a la hora de implementar un sistema de autosensado y autoregulación donde la injerencia humana sea mínima. OpenDDS, siendo una implementación de código abierto del protocolo DDS permitió una solución modular, eficaz y con capacidad de escala a la hora de generar una red de transmisión de información

Con la prueba de concepto hecha, es posible seguir a mejorando el sistema para que cumpla estándares residenciales o industriales donde la capacidad de envío de información y autoregulación puede llegar a ser crucial cuando es utilizado en lugares inhospitos del territorio nacional, liberando así la necesidad de que un o unos individuos estén constantemente atentos al comportamiento a la red de distribución de potencia y abaratar costos en la operación de la red. La implementación de un sistema parecido a este pero no en el contexto de las micro-redes también es posible debido a que la red se trata principalmente de la autoregulación propia ocupando datos de los distintos elementos que la componen.

En resumen, para este proyecto se aprovechó la modularidad que permite OpenDDS, la capacidad de descubrimiento de los distintos elementos de la red, la creación de subscribers y publishers, el tipo de variable de dato a ocupar, todo lo anterior se trabajó de forma independiente y luego se pudo juntar de forma íntegra, lo que permite concluir que un sistema de envío de información y autoregulación es posible con el paradigma publicador y subscriber que ofrece DDS.

## 4.0.3. Trabajo futuro

El sistema funcionó de manera precisa en la mayoría de los casos pero aun así se pudo observar que existen momentos en que la conexión de los entre los publishers y los subscribers se terminaba. Una característica que falta integrar en el sistema es la capacidad de reconexión forzada en conjunto de una espera de datos inteligente. De esta forma aunque existan casos de pérdida de conexión el sistema debiese de forma automática percatarse de esta condición y apresurar la reconexión entre publishers y subscribers.

La conexión de los dispositivos fue mediante una WLAN, la cual es un caso más bien

conveniente y poco cercano a la realidad debido a que, por lo general, los elementos de una red de GTD se encuentran dispersos por una zona física local, haciendo difícil la capacidad de una WLAN. Una posterior mejora es intentar la misma arquitectura pero considerando conexión remota por Internet. Además de la prueba implementada, también uno de los objetivos de este documento es poder ser utilizado como una base para futuros proyectos basados en DDS. La sección 4.0.2 presenta como comenzar en un proyecto parecido al mostrado.

# Bibliografía

- [1] Imagen utilizada de una micro-red. [Online]. Available: [https://www.pewtrusts.org/-/media/post-launch-images/2016/02/brief3\\_figure1v2.jpg?la=en&hash=05123A358E2028BADF9EDC8B7C25C4CF10D5CEB7](https://www.pewtrusts.org/-/media/post-launch-images/2016/02/brief3_figure1v2.jpg?la=en&hash=05123A358E2028BADF9EDC8B7C25C4CF10D5CEB7)
- [2] Laboratorio de micro redes y electromovilidad. [Online]. Available: <http://ingenieria.uchile.cl/investigacion/presentacion/laboratorios/departamento-de-ingenieria-electrica/90784/laboratorio-de-micro-redes-y-electromovilidad>
- [3] Se concreta la interconexión de los sistemas eléctricos sic y sing con la presencia de la presidenta de la república. [Online]. Available: <https://www.cne.cl/prensa/prensa-2017/11-noviembre-2017/>
- [4] R. Ianconescu and V. Vulfin, “Tem transmission line radiation losses analysis,” in *2016 46th European Microwave Conference (EuMC)*. London, UK: IEEE, Oct. 5–9, 2016, pp. 469–472. [Online]. Available: <https://ieeexplore.ieee.org/document/7824381>
- [5] D. T. Ton and M. A. Smith, “The u.s. department of energy’s microgrid initiative,” *The Electricity Journal*, vol. 25, no. 8, pp. 84–94, Oct. 2012.
- [6] A. Hirsch, Y. Parag, and J. Guerrero, “Microgrids: A review of technologies, key drivers, and outstanding issues,” *Renewable and Sustainable Energy Reviews*, vol. 25, pp. 402–411, Jul. 2018.
- [7] T. Porsinger, P. Janik, Z. Leonowicz, and R. Gono, “Component modelling for microgrids,” in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*. Florence, Italy: IEEE, Jun. 7–10, 2016. [Online]. Available: <https://ieeexplore.ieee.org/document/7555869>
- [8] H. B. Cetinkaya, S. Kucuk, M. Unaldi, and G. Berrin, “A case study of a successful industrial microgrid operation,” in *2017 4th International Conference on Electrical and Electronic Engineering (ICEEE)*. Ankara, Turkey: IEEE, Apr. 8–10, 2017. [Online]. Available: <https://ieeexplore.ieee.org/document/7935800>
- [9] Object management group®. [Online]. Available: <https://www.omg.org/>
- [10] Página de descarga de la documentación protocolo rtps. [Online]. Available:

<https://www.omg.org/spec/DDS-RTPS/>

- [11] Página de la guía de desarrollador de opendds. [Online]. Available: <http://download.objectcomputing.com/OpenDDS/OpenDDS-latest.pdf>
- [12] Rti connext dds professional. [Online]. Available: <https://www.rti.com/products/connext-dds-professional>
- [13] Vortex opensplice. [Online]. Available: <http://www.prismtech.com/vortex/vortex-opensplice>
- [14] Opendds. [Online]. Available: <http://opendds.org/>
- [15] Código fuente para opendds. [Online]. Available: <https://github.com/objectcomputing/OpenDDS>
- [16] Uso de debian en el mercado de sistema operativos. [Online]. Available: <https://w3techs.com/technologies/details/os-linux/all/all>
- [17] Página de descarga de raspbian. [Online]. Available: <https://www.raspberrypi.org/downloads/raspbian/>
- [18] Página de descarga de etcher. [Online]. Available: <https://www.balena.io/etcher/>
- [19] Dependencias de opendds. [Online]. Available: <http://opendds.org/documents/building.html>
- [20] Página de descarga de opendds. [Online]. Available: <http://download.objectcomputing.com/OpenDDS/>
- [21] Descarga de herramientas de compilación para arquitecturas tipo arm. [Online]. Available: [download.objectcomputing.com/OpenDDS/resources/gcc-8.2.0-rpi.tar.bz2](http://download.objectcomputing.com/OpenDDS/resources/gcc-8.2.0-rpi.tar.bz2)

# Anexo A

Debido a que uno de los objetivos de este trabajo es implementar una plataforma de transferencia de datos descentralizados para futuros proyectos es que se considera importante explicar y detallar como en esta sección se explicarán los varios tecnicismos existentes. Además de los tecnicismos para poder instalar y operar el sistema mencionado, también se presentan datos y comandos útiles ajenos a este trabajo que le pueden servir al lector.

## 4.1. Sobre el terminal de las distribuciones GNU-Linux

Para el usuario sin experiencia en las distribuciones GNU-Linux, el terminal puede ser considerado una barrera de aprendizaje. Este capítulo de anexos hará uso extensivo del terminal así que será de utilidad referirse a esta sección en particular. Para iniciar el terminal con el teclado se debe hacer la combinación de teclas Ctrl+Alt+t.

### 4.1.1. Sobre *sudo*

Muchas líneas de comandos harán uso de *sudo*, que viene de *substitute user do*. Este comando permite entregarle al usuario privilegios de administrador para comandos que necesiten editar o ingresar información sensible al sistema operativo. Invocado *sudo*, el computador requerirá de una contraseña<sup>1</sup> que el usuario debiese ingresar para adquirir los privilegios de administrador. Notar que al momento de ingresar la contraseña no existe llenado de caracteres, esto es normal y sirve un propósito de seguridad.

### 4.1.2. Comando útiles

Apagado inmediato del computador/dispositivo:

```
$ sudo shutdown -h now
```

Reinicio del computador/dispositivo:

---

<sup>1</sup>La gran mayoría del tiempo, esta contraseña es la misma con la que uno inicia sesión en el computador.

```
$ sudo reboot
```

Manual de información de comando:

```
$ man [comando]
```

Cambiar de directorio:

```
$ cd [directorio]
```

Cambiar al directorio raíz del actual:

```
$ cd ..
```

Listar archivos en un directorio:

```
$ ls
```

Listar todos los archivos (incluyendo los ocultos) en un directorio:

```
$ ls -a
```

Versión de comando (notar que no todos los comandos tienen esta opción):

```
$ [comando] --version
```

Instalación de paquete/software:

```
$ sudo apt-get install [software]
```

Descompresión de archivos con extensión .tar.gz:

```
$ tar xzf [nombre del archivo .tar.gz]
```

Descompresión de archivos con extensión .tar.bz2:

```
$ tar xjf [nombre del archivo .tar.bz2]
```

Borrar archivo:

```
$ rm [archivo]
```

Borrar directorio incluyendo todos sus archivos:

```
$ rm -rf [directorio]
```

Copiar y pegar archivo:

```
$ cp [archivo] [direccion nuevo archivo]
```

Cortar y pegar archivo:

```
$ mv [archivo] [direccion nuevo archivo]
```

Leer/editar archivo en consola (el software *nano* se encuentra instalado de forma automática en Raspbian):

```
$ nano [archivo]
Guardar archivo con Ctrl+O
Cerrar archivo con Ctrl+X
```

## 4.2. Instalación Raspbian

El sistema operativo a elección es Raspbian debido a que es un sistema operativo basado en Debian<sup>2</sup>, implementado específicamente para Raspberry Pi. Para la instalación de este sistema operativo se requiere de una Raspberry Pi<sup>3</sup> y una tarjeta SD<sup>4</sup>.

En un computador estándar (con capacidad para leer tarjetas SD), descargar el sistema operativo desde su página web[17]<sup>5</sup>. También se recomienda descargar Etcher[18], el cual es un programa capaz de montar archivos .iso o .zip en una tarjeta SD. Descargado el sistema operativo es necesario ingresar una tarjeta SD en el computador para luego hacer uso de Etcher. Hecho esto, retirar la tarjeta del computador e ingresarla a la Raspberry Pi, notar que para conveniencia del usuario es importante conectar teclado, mouse y pantalla a la Raspberry Pi, así el proceso de instalación es más expedito. Iniciado el dispositivo se iniciará un proceso de instalación preguntando al usuario distintas configuraciones y preferencias. Lo que es **extremadamente** necesario mencionar es que terminada la instalación, el usuario debe aplicar los siguientes comandos:

```
$ sudo systemctl start ssh
$ sudo systemctl enable ssh
```

El primer comando sirve para habilitar SSH para esta sesión y el segundo para habilitar SSH cada vez que se inicie Raspberry Pi. Terminado esto, reiniciar Raspberry Pi y notar que ya no es necesario ocupar los comandos ingresados anteriormente. Hecho esto ya no es necesario el uso de teclado, mouse y pantalla debido a que se le entregarán comandos a la Raspberry Pi mediante SSH haciendo uso de un computador estándar.

---

<sup>2</sup>Siendo esta distribución basada de GNU-Linux una de las más estables en la actualidad, usado principalmente en servidores web a través del mundo[16].

<sup>3</sup>Se recomienda la versión 3 de este hardware.

<sup>4</sup>Se recomienda una capacidad de almacenamiento de 16GB, clase 10 y de tipo microSD.

<sup>5</sup>Se ofrece más de un tipo de instalación para Raspbian, para este trabajo se utilizó *Raspbian [version-name] with desktop and recommended software*

## 4.3. Instalación de OpenDDS

Una vez instalado Raspbian, es necesario instalar y configurar OpenDDS para su posterior uso. A diferencia de muchos programas que se descargan para la plataforma Windows o incluso las distribuciones basadas en GNU-Linux, es necesario descargar, compilar y luego instalar OpenDDS. Es importante notar que este proceso de instalación se efectúa en un computador estándar con una distribución basada en GNU-Linux<sup>6</sup>, es decir, no en una Raspberry Pi<sup>7</sup>. Los requisitos de compilado son los siguientes:

- Compilador C++. En este caso se considera gcc.
- Perl.
- GNU Make.
- TAO.
- Más de 5GB de espacio en el disco duro.

Notar que no se han definido versiones compatibles con OpenDDS ni paginas de descarga debido principalmente a que la mayoría de los sistemas operativos basados en GNU-Linux ya traen instalados estos programas, con la excepción de TAO, el cual será instalado de forma automática durante la instalación de OpenDDS. Para asegurarse de que las dependencias están instaladas en el sistema operativo se recomienda verificar la versión de cada programa 4.1.2. En caso de dudas con versiones de programas necesarios para la compilación de OpenDDS se recomienda verificar[19]

### 4.3.1. Descarga y compilación de OpenDDS

En un computador estándar se necesita descargar el código fuente de OpenDDS para su posterior compilación, esto se logra en [20] y se recomienda descargar la opción con extensión .tar.gz. Es necesario extraer el archivo descargado, referirse a 4.1.2. Debido a que haremos compilación cruzada, se necesita compiladores que puedan generar objetos y ejecutables que la arquitectura tipo ARM (existente en la Raspberry Pi) pueda interpretar. Las herramientas para lo anterior se pueden descargar en [21] y luego de descargar, descomprimir y enviar al directorio /opt/, los siguientes comandos consiguen lo anterior:

```
$ tar xjf gcc-8.2.0-rpi.tar.bz2
$ mv cross-pi-gcc-8.2.0 /opt/cross-pi-gcc-8.2.0
```

Después, dirigirse a el directorio principal de OpenDDS (notar que en este directorio existe

---

<sup>6</sup>Se recomienda cual sea la última versión de Ubuntu

<sup>7</sup>Para cualquiera le puede parecer extraño tratar de compilar e instalar OpenDDS para un computador estándar si es que se necesita OpenDDS para la Raspberry Pi. Este proceso es conocido como compilación cruzada o *cross-compiling*, donde el trabajo de compilación la hace un computador con más capacidad que la Raspberry Pi y luego el programa ya compilado se instala en la Raspberry Pi.

el archivo *configure*) y ocupar el siguiente comando:

```
$ ./configure --target=linux-cross --target-compiler=/opt/$
cross-pi-gcc-8.2.0/bin/arm-linux-gnueabi-g++
```

Para la no confusión del usuario, el signo \$ después de /opt/ en el comando anterior solo simboliza que el comando es muy largo para caber en una sola línea del documento, es por esto que ese símbolo en particular no existe en realidad. Este comando generará la descarga de ACE+TAO si es que no existe en el computador y preparará todos los archivos para su eventual compilación. Finalizado el anterior proceso, basta con generar el siguiente comando para comenzar el proceso de compilación, el cual puede ser demoroso<sup>8</sup>:

```
$ make
```

### 4.3.2. Conexión SSH

Antes de comenzar la instalación en la Raspberry Pi, es útil generar una conexión haciendo uso de SSH en caso de que la Raspberry Pi no cuente con dispositivos periféricos (mouse, teclado y pantalla). La conexión permitira que desde el computador donde se está generando la compilación se pueda controlar la Raspberry Pi de forma remota (estando ambos dispositivos conectados dentro de una LAN, para facilidad de uso). Una forma de saber que elementos están dentro de una red es ocupando el programa *nmap* que hace una escaneo de los elementos de una red. Es necesario instalar este programa en caso de no tenerlo y luego ejecutarlo de la siguiente manera:

```
$ sudo apt-get install nmap
$ nmap 192.168.0.0/24
```

La salida que presenta *nmap* son todos los elementos de una red y si se configuró de forma correcta la conexión SSH en la Raspberry Pi, se observará que estos elementos dentro de la red tienen el puerto 22 abierto<sup>9</sup>. Una vez identificado que es elemento de la red efectivamente es una Raspberry Pi, es necesario anotar que IP local está asociado con cada dispositivo<sup>10</sup>. Por último, general la conexión SSH entre el computador y la Raspberry Pi sigue el siguiente comando:

```
$ ssh USUARIO@DIRECCIONIP
```

Con USUARIO siendo el nombre del usuario existente dentro de la Raspberry Pi (por lo general es *pi*) y DIRECCIONIP es la dirección IP local encontrada con *nmap*. La primera conexión con el dispositivo generará una llave de seguridad para futuras conexiones con el mismo dispositivo, aquí solo basta con aceptar. Luego se requerirá de una contraseña, la cual es *raspberrypi* si es que el usuario no la cambio con anterioridad. Tanto el nombre de usuario

---

<sup>8</sup>Cuanto se demore el proceso de compilación depende principalmente de la capacidad de procesamiento del computador pero no es inusual un proceso de compilación que dure más de una hora

<sup>9</sup>Este puerto es el que ocupa SSH

<sup>10</sup>Si los dispositivos están dentro de una LAN, el formato viene de la forma 192.168.0.X, con X un número dependiente de como el router local asoció a cada dispositivo dentro de la red.

y contraseña de la Raspberry Pi son configurables.

### 4.3.3. Traspaso de OpenDDS a la Raspberry Pi

Terminada la compilación en el computador estándar, es necesario comprimir el directorio completo OpenDDS-[version] para luego enviar el comprimido a la Raspberry Pi mediante SSH. Una vez terminada la transferencia, descomprimir el archivo. Los comandos necesarios son los siguientes:

```
$ tar czf OpenDDS.tar.gz OpenDDS-[version]
$ scp OpenDDS.tar.gz USER@ADDRESS:[direccion donde se quiere$
dejar el comprimido en la Raspberry Pi]
Los siguiente scomandos son para la Raspberry Pi a traves de SSH
$ cd [direccion donde esta el archivo comprimido]
$ tar xzf OpenDDS.tar.gz
```

Con todo lo anterior OpenDDS ya se encuentra instalado en la Raspberry Pi pero todavía no listo para ocupar debido a que el dispositivo necesita recordar donde se encuentran los directorios de importancia de OpenDDS para su posterior uso. Haciendo uso de la conexión SSH para controlar la Raspberry Pi se debe dirigir al directorio hogar del dispositivo, en este directorio se encontrará el archivo oculto .bashrc, el cual es un archivo que asigna de forma automática variables de entorno del sistema operativo cada vez que inicia la Raspberry Pi. Se debe editar el archivo agregando una línea final con la siguiente información:

```
source [Direccion de directorio OpenDDS]/build/target/setenv.sh
```

El problema que se debe corregir con lo anterior es que el archivo setenv.sh en la dirección especificada anteriormente genera variables de entorno dependiente a direcciones de directorio con respecto al computador donde se generó la compilación. Es necesario entonces editar las direcciones asignadas a las variables de entorno en el archivo setenv.sh:

```
$ cd [Direccion de directorio OpenDDS]/build/target/setenv.sh
$ nano setenv.sh
Editar archivo de la siguiente manera:
export HOST_ACE=[Direccion directorio OpenDDS]/build/host$
/ACE_wrappers
export TAO_ROOT=[Direccion directorio OpenDDS]/build/target/$
ACE_wrappers/TAO
export MPC_ROOT=[Direccion directorio OpenDDS]/ACE_wrappers/MPC
export HOST_DDS=[Direccion directorio OpenDDS]/build/host
export ACE_ROOT=[Direccion directorio OpenDDS]/build/target/$
ACE_wrappers
export DDS_ROOT=[Direccion directorio OpenDDS]/build/target
export DANCE_ROOT=unused
export PATH=${PATH}:[Direccion directorio OpenDDS]/build/host$
/ACE_wrappers/bin:[Direccion directorio OpenDDS]/build/host/bin
```

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$
[Direccion directorio OpenDDS]/build/target/ACE_wrappers/lib:$
[Direccion directorio OpenDDS]/build/target/lib
export CIAO_ROOT=unused
```

Es necesario recordar que los símbolos \$ al final de cada línea solo representan continuación de línea y no existen dentro del archivo a editar. Finalmente, para corroborar los cambios, reiniciar Raspberry Pi y luego en el terminal comprobar que las variables de estado han sido configuradas de manera correcta con:

```
$ echo $DDS_ROOT
```

Si el terminal entrega una dirección correcta entonces se puede asumir que demás variables de estado han sido

#### 4.3.4. Ejemplo de instalación

Terminar si queda tiempo

### 4.4. Sobre el uso de OpenDDS

Con todo el proceso anteriormente se logró instalar OpenDDS tanto en la Raspberry Pi como en el computador estándar donde se generó la compilación. Habiendo dicho lo anterior existen ciertas restricciones que cualquier usuario debe entender si es que hizo uso de este proceso de instalación:

- La Raspberry Pi NO puede compilar y generar ejecutables para OpenDDS pero SI puede correr ejecutables para OpenDDS.
- El computador estándar NO puede correr ejecutables para OpenDDS pero SI puede compilar y generar ejecutables para OpenDDS

Como se puede ver, se recomienda que la metodología de trabajo considere escribir código, compilar y luego generar ejecutables en el computador estándar para posteriormente enviar estos ejecutables y todas las librerías de las que los ejecutables dependen a la Raspberry Pi<sup>11</sup>. Esto es una consecuencia de ocupar compilación cruzada.

---

<sup>11</sup>Mediante SSH, por ejemplo