



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

Titulació:

Master Universitario en Ingeniería de Sistemas Automáticos y
Electrónica Industrial

Alumno:

Diego Andrés Lascano Dalgo

Enunciado del Trabajo de Fin de Master:

Proyecto de una aplicación HEMS para la gestión de energía de hogares con generación de energía fotovoltaica situados en Ecuador.

Director TFM:

Joan Ferran Salaet Pereira

Convocatoria de entrega:

Junio 2019

AGRADECIMIENTOS

Quiero aprovechar este espacio para agradecer a la Secretaria Nacional de Educación Superior, Ciencia y Tecnología, SENESCYT, y al Instituto del Fomento al Talento Humano, IFTH, porque mediante el programa de becas para estudios de cuarto nivel, se me ha permitido continuar con mi formación académica y de esta manera estar mejor preparado para aportar en el desarrollo de mi país Ecuador.

Además, quiero agradecer de manera especial a mis padres Marcelo y Marcia, quienes han guiado mi camino para ser la persona que ahora soy, siendo parte de los logros conseguidos. A mis hermanos Cristhian y Ronald por ser mis mejores amigos siempre que los necesité, y a mi esposa Valeria por estar a mi lado y darme todo su apoyo incondicional.

RESUMEN

Este documento representa la memoria del Trabajo de Fin de Máster del programa de estudio Master Universitario en Ingeniería de Sistemas Automáticos y Electrónica Industrial, de la Universidad Politécnica de Catalunya (UPC).

El presente proyecto trata sobre el diseño y desarrollo de una aplicación multiplataforma, para la gestión de la energía eléctrica consumida en los hogares, incluyendo generación de energía fotovoltaica.

En los primeros capítulos se puede encontrar la parte introductoria donde se detallan los objetivos y especificaciones del estudio, además del alcance establecido. También, en esta sección se presenta el estado del arte, que incluye un análisis completo de la gestión de energía en los hogares en la actualidad, considerando investigaciones científicas publicadas en relación a este tema.

Luego, se realiza el análisis y selección de las herramientas necesarias para poder desarrollar la aplicación considerando aspectos tanto técnicos como económicos.

Una vez definidas las especificaciones y herramientas necesarias para la elaboración de la aplicación, se presenta el desarrollo de cada componente del sistema empezando por el diseño del sistema de generación fotovoltaica, seguido por el modelamiento matemático del sistema hasta llegar al desarrollo del algoritmo que genera cronogramas de funcionamiento óptimo para los artefactos automáticos del hogar.

Finalmente, se presenta la aplicación multiplataforma desarrollada y su interfaz de usuario. Se detallan sus principales funcionalidades y como el usuario puede gestionar su propia instalación. Asimismo, se presentan las herramientas de análisis de las curvas que se generan gracias a los cronogramas de optimización.

ABSTRACT

This document represents the report of the Master's Final Project for the Master in Automatic Systems and Industrial Electronics program of the Polytechnic University of Catalonia (UPC).

This project is about the design and development of a multi-platform application, meant to manage the electric energy consumed by households, including photovoltaic energy generation

The introductory part of this project can be found in the first chapters, where the objectives and specifications are detailed. Likewise, in this section is presented the state-of-the-art, which includes a complete analysis of the current home energy management systems, considering published scientific researches related to this topic.

Next, in order to properly select the tools needed to develop the application, an analysis is performed to evaluate the options considering both technical and economic aspects.

Once the technical specifications and the tools needed to develop the application are defined, the actual development of each component of the system is detailed, starting with the design of the photovoltaic generation system, followed by the mathematical modeling of the entire household, until the development of the algorithm that generates optimal operation schedules for the automatic operated home applications.

Finally, the multi-platform application is presented, along with the user interface designed. Its main functionalities are detailed, and how the user can manage their own installation is explained as well. Moreover, the tools for the correct analysis of the results achieved due to the optimization schedules are shown.

ÍNDICE GENERAL

AGRADECIMIENTOS	I
RESUMEN	II
ABSTRACT	III
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	IX
LISTA DE ACRÓNIMOS Y TÉRMINOS ESPECÍFICOS	X
1. INTRODUCCIÓN	1
1.1. Objeto del proyecto.	1
1.2. Objetivo	1
1.3. Alcance.	1
1.4. Especificaciones.	2
1.5. Justificación del proyecto.	2
2. ESTADO DEL ARTE	4
2.1. Antecedentes.	4
2.1.1. Tecnología en la energía de los hogares.	7
2.1.2. Sistemas de gestión de energía en el hogar <i>HEMS</i>	9
2.2. Sistemas HEMS en la actualidad.	10
2.3. Investigaciones más recientes	12
3. ANÁLISIS Y SELECCIÓN DE HERRAMIENTAS	14
3.1. Sistema operativo.	14
3.1.1. Windows	14
3.1.2. Linux	14
3.1.3. macOS	15
3.2. Entorno de desarrollo.	15
3.2.1. Laravel	15
3.2.2. LabVIEW	16
3.2.3. Matlab	17
3.2.4. Visual Studio	17
3.3. Algoritmo de optimización.	18

3.3.1.	Algoritmo Genético _____	18
3.3.2.	Optimización por enjambre de partículas _____	20
3.3.3.	Colonia Artificial de Abejas _____	22
3.4.	Selección de herramientas _____	23
3.4.1.	Criterios de selección. _____	23
3.4.2.	Herramientas seleccionadas. _____	24
4.	DESARROLLO DE LA APLICACIÓN HEMS _____	26
4.1.	Sistema de generación fotovoltaica. _____	26
4.1.1.	Condiciones de instalación _____	26
4.1.2.	Diseño de la instalación _____	27
4.1.3.	Simulación de la instalación _____	29
4.2.	Costo dinámico de la energía eléctrica. _____	31
4.3.	Modelamiento del sistema. _____	32
4.3.1.	Definiciones iniciales _____	32
4.3.2.	Formulación del problema. _____	33
4.4.	Aplicación HEMS. _____	37
4.4.1.	Configuraciones básicas de Laravel _____	39
4.4.2.	Algoritmo Genético _____	40
4.4.3.	Parámetros de funcionamiento. _____	49
5.	INTERFAZ DE USUARIO _____	50
5.1.	Funciones básicas. _____	50
5.1.1.	Ingreso de usuario _____	50
5.1.2.	Pantallas de funcionamiento. _____	51
5.2.	Artefactos automáticos del hogar (AOA). _____	54
5.3.	Cronogramas de optimización. _____	56
6.	PRUEBAS Y RESULTADOS _____	57
6.1.	Pruebas. _____	57
6.1.1.	Nuevo cronograma _____	57
6.2.	Análisis de resultados. _____	59
7.	CONCLUSIÓN _____	63
8.	PRESUPUESTO _____	64
9.	TRABAJO FUTURO _____	65

10. BIBLIOGRAFÍA	66
ANEXOS	69
ANEXO A – Valor Técnico Ponderado	70
ANEXO B – Hoja de datos del panel solar SPR-X21-335-BLK	72
ANEXO C – System Advisor Model	73

ÍNDICE DE FIGURAS

Figura 2.1. Tendencia de consumo de energía a nivel mundial, según su fuente de producción. _____	5
Figura 2.2. Evolución de clientes por grupo de consumo. _____	5
Figura 2.3. Evolución de la energía utilizada por grupo de consumo. _____	6
Figura 2.4. Evolución histórica y proyección del consumo energético en el sector residencial. _____	6
Figura 2.5. Interés a lo largo del tiempo de acuerdo con Google Trends sobre IoT, Smart Home, y Smart Grid. _____	8
Figura 2.6. Arquitectura típica de un HEMS. _____	9
Figura 2.7. CT200i Smart Hub de la marca carbonTRACK® _____	10
Figura 2.8. Dashboard web de Smappee. _____	11
Figura 3.1. Partes del algoritmo genético. _____	19
Figura 3.2. Operaciones del algoritmo genético. _____	19
Figura 3.3. Diagrama de flujo del algoritmo genético. _____	20
Figura 3.4. Diagrama de flujo del algoritmo PSO. _____	21
Figura 3.5. Algoritmo del método ABC. _____	22
Figura 4.1. Tejado disponible para la instalación de los paneles de generación PV. _____	26
Figura 4.2. Panel solar SPR-X21-335-BLK de la marca Sun Power _____	28
Figura 4.3. Layout de la instalación de generación fotovoltaica. _____	28
Figura 4.4. Inversor PS247-05-180 _____	29
Figura 4.5. Perfil de generación fotovoltaica estimada, para 3 días específicos. _____	30
Figura 4.6. Perfiles de potencia generada del día característico de cada mes, de un año meteorológico típico. _____	30
Figura 4.7. Costo horario de la energía. Tarifa 2.0 DHA _____	31
Figura 4.8. Inicio de operación OST: (a) inicio al comienzo del periodo (b) inicio al final del periodo. _____	33
Figura 4.9. Relación de los parámetros de configuración de los AOA. _____	34
Figura 4.10. Definición del DTR. _____	35
Figura 4.11. Paradigma Modelo-Vista-Controlador. _____	37
Figura 4.12. Directorio raíz del proyecto Homenergymanager. _____	38
Figura 4.13. Directorio de la carpeta Controllers. _____	38
Figura 4.14. Directorio de las carpetas js, sass y views. _____	39
Figura 4.15. Ventana de creación de base de datos en phpMyAdmin. _____	39

Figura 4.16. Configuración de la base de datos en el archivo .env _____	40
Figura 4.17. Directorio del algoritmo genético. _____	41
Figura 4.18. Cabecera de la clase SchedulingGA. _____	44
Figura 4.19. Diagrama de flujo ideal de un sistema ADR (automated demand response). _____	45
Figura 4.20. Diagrama de flujo del algoritmo genético. _____	47
Figura 4.21. Diagrama de flujo del subproceso “Evaluar Población”. _____	48
Figura 5.1. Ventana de bienvenida a la aplicación HEMS. _____	50
Figura 5.2. Ventana de registro de nuevo usuario. _____	51
Figura 5.3. Ventana de ingreso de usuario previamente registrado. _____	51
Figura 5.4. Pantalla de inicio de la aplicación. _____	52
Figura 5.5. Despliegue de datos de las ventanas Histórico y Hoy. _____	53
Figura 5.6. Tarjeta de resumen diario. _____	53
Figura 5.7. Ventana Gráficas, para comparaciones de variables en la base de datos. _____	54
Figura 5.8. Ventana Mi instalación. AOA dentro de la instalación del usuario. ____	54
Figura 5.9. Ventana de ingreso de un nuevo AOA. _____	55
Figura 5.10. Ventana Mi instalación con los artefactos automáticos registrados. _	55
Figura 5.11. Ventana de acciones de control para los AOA. _____	56
Figura 5.12. Ventana Mis cronogramas con los cronogramas generados por el usuario actual. _____	56
Figura 6.1. Evolución de la generación del cronograma en el servidor. _____	57
Figura 6.2. Notificación de nuevo cronograma en proceso. _____	58
Figura 6.3. Generación de nuevo cronograma finalizado exitosamente. _____	58
Figura 6.4. Generación del cronograma en el servidor finalizado exitosamente. __	59
Figura 6.5. Resultados del consumo energético del día en términos de coste. ____	60
Figura 6.6. Tendencia de costo de la energía consumida para el día 8 de junio __	61
Figura 6.7. Resultados del consumo energético del día en términos de energía. _	61
Figura 6.8. Tendencia de la energía consumida para el día 8 de junio. _____	62

ÍNDICE DE TABLAS

Tabla 2.1. Ecuador. Facturación de energía a nivel nacional; clientes regulados por pliegos tarifarios. _____	7
Tabla 3.1. Criterios de selección del sistema operativo. _____	23
Tabla 3.2. Criterios de selección del entorno de desarrollo. _____	23
Tabla 3.3. Criterios de selección del algoritmo de optimización. _____	24
Tabla 3.4. VTP para el sistema operativo. _____	24
Tabla 3.5. VTP para el entorno de desarrollo. _____	24
Tabla 3.6. VTP para el algoritmo de optimización. _____	25
Tabla 4.1. Condiciones para la instalación de la generación fotovoltaica. _____	27
Tabla 4.2. Características del panel solar SUNPOWER SPR-X21-335-BLK _____	27
Tabla 4.3. Características técnicas del inversor POWER STATION PS247-05-180	29
Tabla 4.4. Propiedades y métodos de la Clase “Individual”. _____	41
Tabla 4.5. Propiedades y métodos de la Clase “Population”. _____	42
Tabla 4.6. Métodos y propiedades de la Clase “Schedule”. _____	42
Tabla 4.7. Propiedades y métodos de la Clase “GeneticAlgorithm”. _____	43
Tabla 4.8. Propiedades y métodos de la Clase “SchedulingGA” _____	44
Tabla 4.9. Parámetros de funcionamiento del HEMS y Algoritmo Genético. _____	49
Tabla 4.10. Artefactos automáticos AOA y sus preferencias de usuario. _____	49
Tabla 6.1. Preferencias ingresadas por el usuario y propuestas por el cronograma óptimo. _____	59
Tabla 8.1. Presupuesto general del proyecto. _____	64

LISTA DE ACRÓNIMOS Y TÉRMINOS ESPECÍFICOS

HEMS:	Home Energy Management System.
BACK-END:	Capa de acceso de datos.
FRONT-END:	Capa de presentación de datos.
CONALEC:	Consejo Nacional de Electricidad del Ecuador.
PSO:	Optimización por enjambre de partículas (Particle Swarm Optimization)
ABC:	Colonia de abejas artificial (Artificial Bees Colony)
ORM:	Mapeo de objeto relacional (Object-Relational Mapping)
FERUM:	Fondo de electrificación rural urbano marginal.
ERES FERUM:	Energía eléctrica consumida Residencial pertenecientes al FERUM
ERES S FERUM:	Energía eléctrica consumida Residencial no pertenecientes al FERUM
AOA:	Artefactos operados automáticamente (Automatic-Operated Appliance)
MOA:	Artefactos operados manualmente (Manually Operated Appliance)
RTP:	Precio en tiempo real (real-time pricing)
OST:	Periodo de tiempo de inicio de operación (Operation Start time slot)
OTI:	Intervalo de tiempo para la operación (Operation Time Interval)
LOT:	Duración de operación (Length of Operation Time)
timeslot:	Periodo de tiempo mínimo de 12 minutos.
u:	Periodo de tiempo (Time slot)

1. INTRODUCCIÓN

En el primer capítulo del trabajo se presentan las condiciones iniciales donde se da a conocer el objeto del proyecto, los objetivos planteados, el alcance, las especificaciones o requerimientos básicos para el desarrollo, y la justificación del porque se lleva a cabo este presente trabajo.

1.1. Objeto del proyecto.

El objeto del proyecto es diseñar e implementar una aplicación para la gestión de la energía eléctrica en los hogares, considerando además una instalación de generación de energía fotovoltaica.

1.2. Objetivo

El objetivo de este proyecto es crear una aplicación multi plataforma que sea capaz de administrar la energía del hogar generando cronogramas de funcionamiento para los artefactos automáticos en los hogares, con la finalidad de optimizar el consumo de energía eléctrica lo que a su vez reduciría el costo de la energía facturada durante el día.

1.3. Alcance.

El alcance del presente proyecto comprende las siguientes actividades:

- Modelamiento matemático de las cargas involucradas en un hogar para obtener la función de coste a optimizar.
- Creación de una base de datos con la energía fotovoltaica generada durante un año meteorológico típico en una zona específica del Ecuador, con una frecuencia horaria. Se utilizará la herramienta/software *System Advisor Model (SAM)*.
- Generación de una base de datos con el costo de la energía eléctrica del Ecuador para un año, con periodos de actualización de una hora entre cada dato.
- Selección del entorno de desarrollo más adecuada para construir la aplicación HEMS.
- Selección y programación de los algoritmos necesarios para generar la planificación de funcionamiento más óptimos que permitan minimizar el costo de la energía facturada.
- Diseño y programación de la interfaz gráfica para la interacción de los usuarios con la aplicación HEMS.
- Generación de una herramienta que permita el análisis de los datos entregados por la aplicación HEMS.
- Consolidar una aplicación HEMS funcional que pueda ser utilizada en múltiples plataformas.
- La aplicación HEMS desarrollada solo comprende la parte de software. No se considera parte del alcance la integración con dispositivos físicos tales como

interruptores o medidores inteligentes, dejando estas tareas para futuras investigaciones.

1.4. Especificaciones.

Se han definido las siguientes especificaciones básicas para el desarrollo del presente proyecto:

- La aplicación HEMS debe ser intuitiva y fácil de manejar, ya que el usuario final será una persona dentro de un hogar, la misma que probablemente cuente con conocimientos básicos de tecnología.
- Los datos usados durante el desarrollo deben ser datos históricos reales.
- Se dará preferencia al uso de herramientas/software libre durante el desarrollo de la aplicación HEMS.

1.5. Justificación del proyecto.

El consumo energético se ha convertido en una preocupación a nivel mundial. A medida que los años pasan, el número de habitantes en las grandes urbes crece, a la vez que los estándares de vida en los países en vías de desarrollo son cada vez mejores, la demanda de energía ha ido en constante aumento debido a la necesidad de utilizar cada vez más artefactos y dispositivos que mejoran la calidad de vida de los seres humanos, o a su vez ayudan o facilitan sus tareas en su diario vivir. Un ejemplo de esto es la climatización de ambientes de trabajo, que, dependiendo de las condiciones climáticas, requieren más o menos energía para satisfacer al usuario. Otro ejemplo importante es el consumo de información mediante dispositivos móviles ya que, por una parte, involucra el consumo de energía de los dispositivos per se, mientras que, por otro lado, se genera una necesidad aun mayor de mantener servidores, antenas repetidoras y demás accesorios que generan las redes de datos encendidos de forma ininterrumpida alrededor del mundo. En consecuencia a esta problemática, se ha generado un nuevo ámbito de estudio enfocado a mejorar el uso de energía en los lugares donde se concentra el consumo de energía como los hogares.

El consumo de energía en los hogares es uno de los temas más discutidos hoy en día. Ciudades con alta densidad poblacional requieren cada vez más fuentes de generación de energía para poder abastecer a sus hogares, mientras que buscan concientizar a la población sobre el uso adecuado de la misma, en especial durante las horas pico del día donde se puede evidenciar los altos valores de energía consumida.

Considerando el panorama de la problemática vista desde el lado del usuario existen dos puntos importantes que analizar:

Por una parte, existe la necesidad de los usuarios de llevar a cabo sus tareas del hogar durante el tiempo que ellos tienen libre. Tareas como lavar ropa, lavar platos o usar la aspiradora para limpiar el piso son generalmente realizadas cuando los usuarios regresan a sus hogares después de la jornada laboral, lo cual desencadena en una alta

demanda energética prácticamente coordinada en franjas horarias muy marcadas, que son las zonas de principal interés.

Por otra parte, está presente el costo asociado a la energía consumida por los mismos usuarios. Este costo de energía introduce un nuevo factor al sistema, donde se puede ver claramente que el precio está estrechamente relacionado, de forma directamente proporcional, a la demanda instantánea; esto quiere decir que, las mismas horas de alta demanda de energía antes mencionadas, conllevan a un valor mayor en la facturación de este servicio.

Por esta razón, es de vital importancia llevar a cabo estudios y también desarrollar herramientas que puedan ayudar a los usuarios, a planificar y gestionar de forma eficiente el uso de la energía en sus hogares tomando en cuenta principalmente los dos puntos antes mencionados: la satisfacción del cliente al realizar sus actividades lo antes posible, y la reducción del consumo en horas pico que generen un decremento en los valores a pagar por el servicio. De esta forma se estaría mitigando de forma considerable la problemática de demanda energética a nivel mundial, que de seguro seguirá creciendo.

2. ESTADO DEL ARTE

En este capítulo se presentan los antecedentes referentes al tema de estudio. Además, se analizan las aplicaciones de gestión de energía en los hogares existentes en el mercado actual y los métodos de optimización aplicables a la problemática de planificación de tareas.

2.1. Antecedentes.

Para tener una visión más clara de la problemática a la que se enfrentan los hogares a nivel mundial, es importante recabar información histórica sobre cómo ha ido evolucionando la demanda energética, en especial en el sector domiciliario.

Según BP, anteriormente conocida como *British Petroleum*, una empresa dedicada a la energía, indica en su revista *Statistical Review of World Energy* (BP, 2018) que en el año 2017, el consumo de energía ha crecido en un 2.2% respecto al año 2016, que a su vez vio un incremento del 1.2% respecto al 2015.

En la Figura 2.1 extraída de la revista (BP, 2018), se puede observar una gráfica que indica la tendencia de consumo de energía a nivel mundial, identificando el tipo de fuente con su respectiva proporción. De esta se puede destacar que después de una pequeña depresión aproximadamente en el año 2009, el consumo continúa su marcado crecimiento con una pendiente incluso más elevada que décadas anteriores. Asimismo, es importante mencionar la evolución del uso de las energías renovables, que durante aproximadamente los últimos 7 años han mostrado una tasa de crecimiento destacable frente a otras fuentes de generación de energía, a pesar de que su participación global respecto a las demás es muy pequeña.

Esta tendencia de consumo energético a nivel mundial evidenciada a lo largo de los años, corresponde a las actividades desarrolladas por el ser humano en los diferentes sectores como el industrial, residencial, comercial, entre otros. Cada sector se caracteriza por sus valores específicos como número de clientes, potencia eléctrica consumida y los horarios de funcionamiento más significativos.

World consumption
Million tonnes oil equivalent

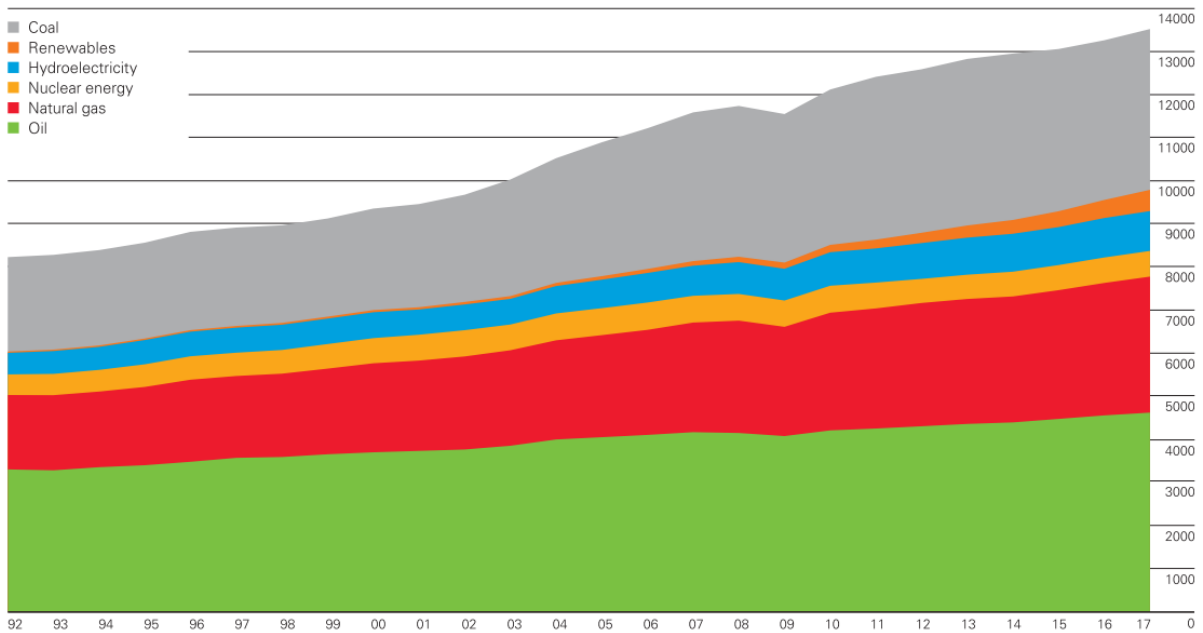


Figura 2.1. Tendencia de consumo de energía a nivel mundial, según su fuente de producción.

Ahora bien, si se considera el sector residencial, el cual es objeto del presente proyecto, se pueden extraer características muy puntuales que definen la situación actual dentro de la problemática energética. Considerando el caso particular del Ecuador, un país en vías de desarrollo situado en América del Sur, en el documento *Plan Maestro de Electrificación del Ecuador* (CONELEC, 2013), se muestra de forma detallada el análisis sobre la demanda energética del país, destacando los diferentes sectores consumidores del servicio.

La Figura 2.2 extraída de (CONELEC, 2013), muestra cómo ha evolucionado el número de clientes en cada sector. De esta se puede extraer como punto importante el crecimiento del sector residencial desde el año 2003 hasta el 2012, donde el número de clientes consumidores del servicio de energía eléctrica ha incrementado en un 57%.

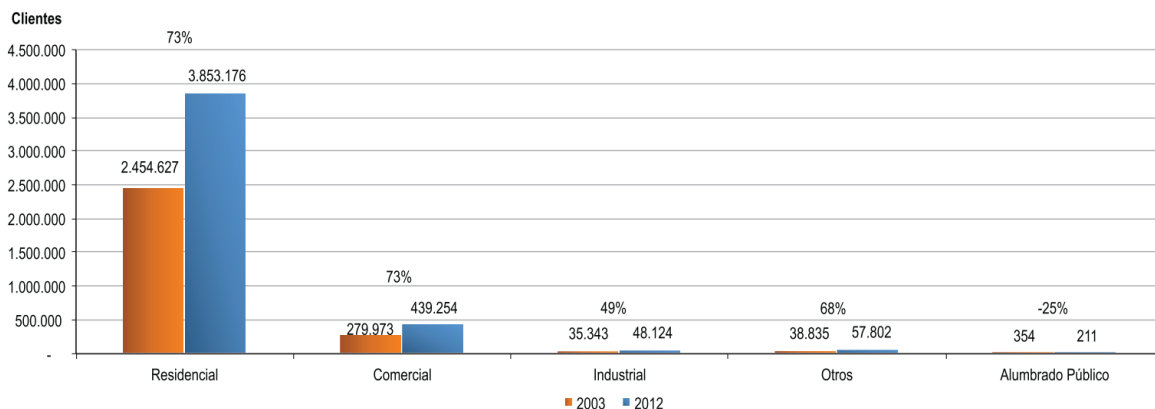


Figura 2.2. Evolución de clientes por grupo de consumo.

Del mismo modo, durante este periodo, también se presenta el incremento en términos de energía eléctrica utilizada. Esto se puede ver en la Figura 2.3 obtenida de (CONELEC, 2013)

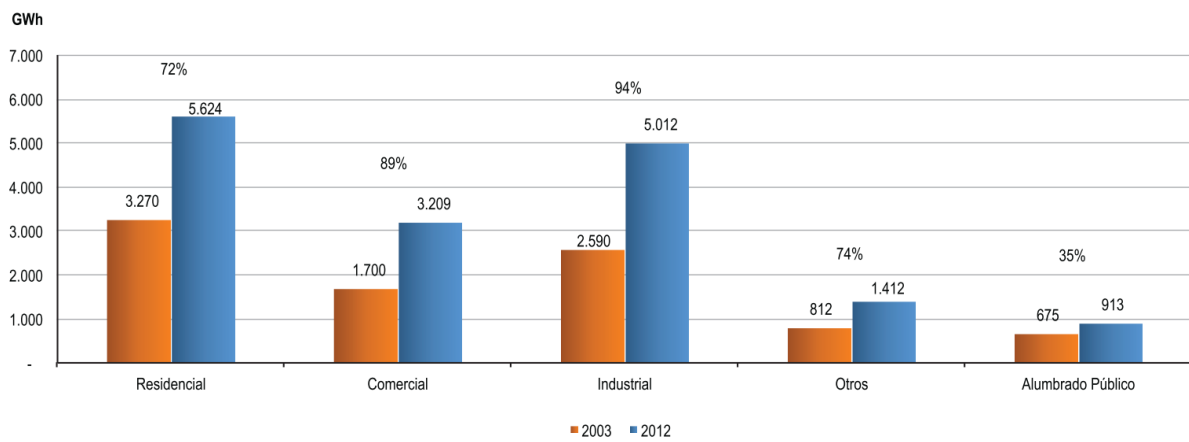


Figura 2.3. Evolución de la energía utilizada por grupo de consumo.

Analizando la Figura 2.3, se puede destacar que el incremento del consumo de energía en el sector residencial, alcanza un valor del 72%, siendo el tercer sector con mayor crecimiento de demanda de energía, debajo de los sectores industrial y comercial.

Finalmente, considerando todos los datos expuestos anteriormente, en este análisis de demanda energética (CONELEC, 2013), mediante la Figura 2.4 se presenta una proyección hasta el año 2022 como horizonte de estudio.

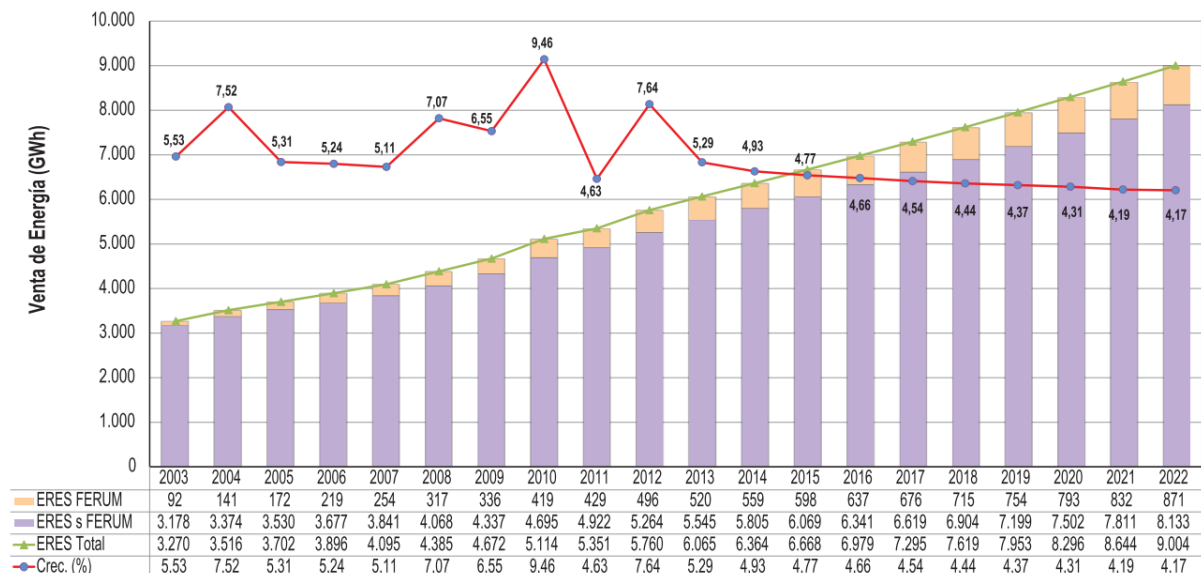


Figura 2.4. Evolución histórica y proyección del consumo energético en el sector residencial.

En la Figura 2.4 se presentan dos puntos importantes que sustentan la realidad y el futuro cercano respecto al consumo de energía en el sector residencial en el Ecuador. Por una parte, mediante una línea roja sobre la gráfica se puede observar la tasa de crecimiento poblacional del país donde resalta el hecho de que siempre hay un valor positivo, es decir, en todos los años se prevé un aumento en la población. En cambio,

mediante una línea de color verde, se indica la proyección de crecimiento de consumo de energía del sector residencial, el mismo que alcanza un máximo de 9000 *GWh* en el año 2022. Este valor representa un incremento del 56% respecto al inicio del periodo de estudio en el año 2012. Cabe destacar que, en el periodo de estudio anterior, desde el año 2003 hasta el 2012, en el sector residencial se determinó un crecimiento del 57% el cual es muy próximo al valor proyectado hasta el año 2022.

Con la finalidad de comparar los valores proyectados, se han contrastado con datos entregados en el reporte *Energía eléctrica y su incidencia en las actividades productivas* (Guayaquil, 2017) que se muestran en la Tabla 2.1.

En dicha tabla se pueden ver los valores de la energía facturada en los diferentes sectores. Cabe resaltar que en lo que respecta al sector residencial, los valores proyectados muestran un error relativo no mayor al 3% con lo cual se puede aceptar los demás valores proyectados.

Tabla 2.1. Ecuador. Facturación de energía a nivel nacional; clientes regulados por pliegos tarifarios.

	2007 [GWh]	2014 [GWh]	2015 [GWh]	2016 [GWh]	2017* [GWh]
Residencial	4.095	6.364	6.928	7.105	6.681
Industrial	1.782	4.975	4.973	4.778	4.504
Comercial	2.205	3.786	3.981	3.838	3.517
Otros	1.217	1.811	1.980	2.049	1.956
Alumbrado Público	765	1.023	1.081	1.127	1.109
TOTAL	10.064	17.958	18.943	18.897	17.767

* valores enero - noviembre

Se pueden resumir los datos analizados en que, tanto en Ecuador como a nivel mundial, al tener un crecimiento poblacional exponencial, se genera un aumento proporcional en el consumo energético, lo cual se convierte en un problema actual y futuro. De esta realidad surge la necesidad de generar nuevas formas o métodos de optimización del uso de los recursos energéticos incluyendo nuevas tecnologías como *Smart Grids*, técnicas de optimización y planificación de consumos energéticos en los hogares, entre otras.

2.1.1. Tecnología en la energía de los hogares.

En los últimos años, las redes inteligentes o *Smart Grids* han venido destacando debido a sus potenciales beneficios como seguridad, eficiencia y en especial su ayuda al medio ambiente, ya que consideran la generación mediante fuentes de energía renovables (Shen, Jiang, Liu, & Wang, 2016). En el futuro, la distribución de la energía hacia los hogares se realizará mediante estas redes, permitiendo el manejo más óptimo de todos los recursos involucrados en el proceso tales como artefactos de casa automáticos y manuales, generación de energía fotovoltaica y eólica, medidores de energía inteligentes, entre otros.

Un aspecto fundamental en el desarrollo de las redes inteligentes es la comunicación entre dispositivos. La combinación de las tecnologías de la información junto con

nuevas tecnologías de comunicación y sistemas de medición, han dado lugar a un nuevo concepto que, aunque es aplicado a más industrias, juega un papel fundamental en los sistemas HEMS; este es el internet de las cosas, de sus siglas en inglés *IoT* (*Internet of Things*). Gubbi define el *IoT* como una interconexión de dispositivos de medición y actuación que provee la capacidad de compartir información a través de plataformas mediante un entorno de trabajo unificado, desarrollando una imagen operativa común que posibilita aplicaciones innovadoras (Risteska Stojkoska & Trivodaliev, 2017).

De acuerdo a las estadísticas encontradas en *Google Trends* que se muestran en la Figura 2.5, en los últimos 5 años ha existido un incremento pronunciado en el interés respecto al *IoT* y también sobre *Smart Home*.

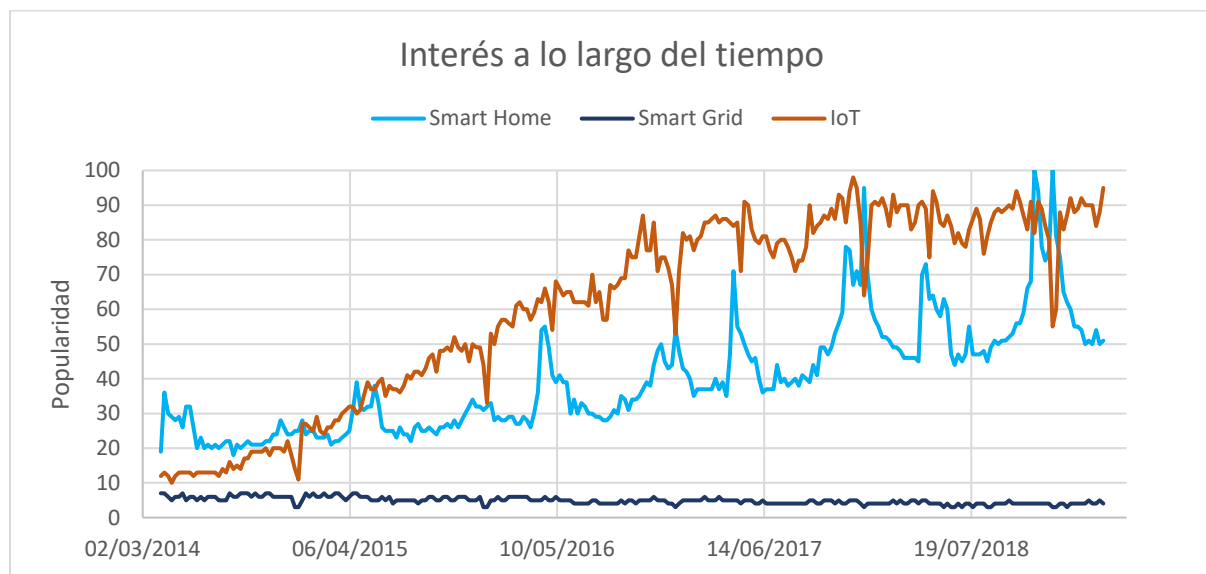


Figura 2.5. Interés a lo largo del tiempo de acuerdo con *Google Trends* sobre *IoT*, *Smart Home*, y *Smart Grid*.

De la Figura 2.5 además se puede extraer que la tendencia apunta a que, en un futuro, estas tecnologías estarán presentes en todas las partes de los ecosistemas tecnológicos de las personas. Prueba de esto son un sin número de productos que cada día son presentados a los usuarios; productos capaces de preparar comida, encender o apagar artefactos eléctricos, o controlar la temperatura de una habitación desde la comodidad de los dispositivos móviles de los usuarios.

Dentro de las redes inteligentes se sitúan las micro redes. Estas, consisten de todas las partes fundamentales en una red de distribución, pero a una escala menor, es decir, constan de una propia fuente de energía como micro turbinas, aerogeneradores o paneles solares fotovoltaicos; también cuentan dispositivos de almacenamiento como baterías y sus propias cargas. A pesar de tener una arquitectura bastante parecida a las tradicionales, gracias a las bondades de los avances tecnológicos como el internet de las cosas *IoT*, las micro redes han logrado interconectarse con el nivel más alto de las redes de distribución de energía, siendo capaces de intercambiar información tal como la variación del costo de la energía en tiempo real, entre otra información relevante (Shen et al., 2016). De esta manera, las micro redes en conjunto con las tecnologías de la información más avanzadas, permiten un mejor manejo de los recursos energéticos, con

respuestas que se ajustan a las necesidades y requerimientos instantáneos, aumentando la eficiencia energética de las instalaciones actuales.

2.1.2. Sistemas de gestión de energía en el hogar *HEMS*

Dentro del concepto de micro redes, surge la necesidad de estudiar los hogares como partes activas y fundamentales en la distribución de energía. Un sistema de gestión de energía en el hogar, de sus siglas en inglés HEMS (*Home Energy Management System*) está compuesto por varias partes como medidores inteligentes, artefactos, medidores de condiciones climáticas, entre otros. En la Figura 2.6 obtenida de (Shareef, Ahmed, Mohamed, & Al Hassan, 2018) se puede observar un esquema general de este tipo de sistemas:

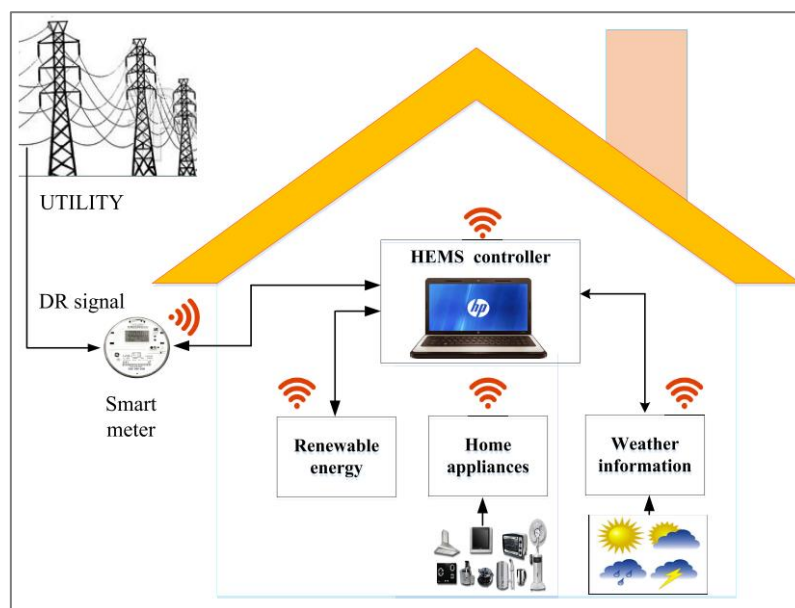


Figura 2.6. Arquitectura típica de un HEMS.

Uno de los puntos más importantes que se debe destacar es que todos los elementos de un HEMS tienen la capacidad de compartir información de forma inalámbrica. Como se puede observar en la Figura 2.6, la arquitectura muestra dispositivos inteligentes de medición, que intercambian información en tiempo real con la red de energía. También se encuentran presentes las fuentes de energía renovable tales como fotovoltaica y eólica, los artefactos propios del hogar como refrigeradora, lavadora, etc., la información climática actual, que influye en el desempeño del sistema, y finalmente, el corazón del sistema que es el controlador HEMS (Shareef et al., 2018).

El objetivo del sistema HEMS es el de manejar de manera automática y eficiente todos los recursos involucrados, de tal forma que se maximice el uso de la energía y a su vez, se pueda planificar el uso de los artefactos para minimizar el costo de energía facturada. Respecto a esto, es importante puntualizar que, en la actualidad, las investigaciones apuntan hacia la búsqueda de nuevas metodologías que permitan generar cronogramas de uso para los artefactos automáticos en el hogar de forma que se pueda reducir la

demanda energética en horas pico, moviendo la mayoría de consumos hacia horarios donde el costo de la energía sea menor. Para ello, es de vital importancia conseguir una buena integración de las diferentes tecnologías involucradas como la generación de energías renovables y la adquisición de información climática en conjunto con un controlador capaz de administrar o gestionar los dispositivos, tomando las decisiones necesarias y considerando la variación del costo de la energía de red.

2.2. Sistemas HEMS en la actualidad.

Actualmente existen varios productos en el mercado dentro de la categoría HEMS. La mayoría de ellos ofrecen características similares entre las cuales destacan la monitorización y análisis de la energía del hogar. Otros, ofrecen la integración y optimización con instalaciones de energía solar fotovoltaica, además de la opción de poder controlar de forma manual, pero remota, los dispositivos del hogar.

En general, características indispensables en estos sistemas comprenden de una herramienta de análisis de la energía, alarmas para eventos dentro del funcionamiento esperado, todo mediante una interfaz gráfica amigable con el usuario, misma que a su vez pueda ser visualizada y utilizada en dispositivos móviles.

Un producto que puede ser encontrado mediante una búsqueda de internet es *carbonTRACK*[®], el cual es de origen australiano. Después de analizar las características que ofrece este sistema, se puede decir que es uno de los productos más completos del mercado, y que es de los pocos que mencionan la planificación automática de tareas (*appliances scheduling*), aunque no se describe a detalle cómo funciona.

Entre las características principales que se ofertan en esta aplicación se encuentran las siguientes:

- Monitorización de dispositivos.
- Planificación de uso de dispositivos (si está disponible).
- Maximización del consumo de energía solar.
- Optimización de energía de baterías.
- Sistema de alarmas de funcionamiento.

Esta marca integra su software dentro de un controlador dedicado, al cual lo han denominado CT200i Smart Hub (Figura 2.7). Este trabaja con los protocolos de comunicación *ZigBee HA 1.2* y *Z-Wave* para su conexión con los diferentes sensores.



Figura 2.7. CT200i Smart Hub de la marca carbonTRACK[®]

Otra marca relacionada con los sistemas de gestión de la energía en el hogar es *Smappee*. En su portal web, esta empresa de origen belga muestra un *dashboard* o tablero de control donde se presenta la información obtenida en tiempo real de los consumos del hogar (Figura 2.8). Esta herramienta basada en la web permite el análisis gráfico o numérico, además de dar seguimiento de los datos generados por los dispositivos del hogar.

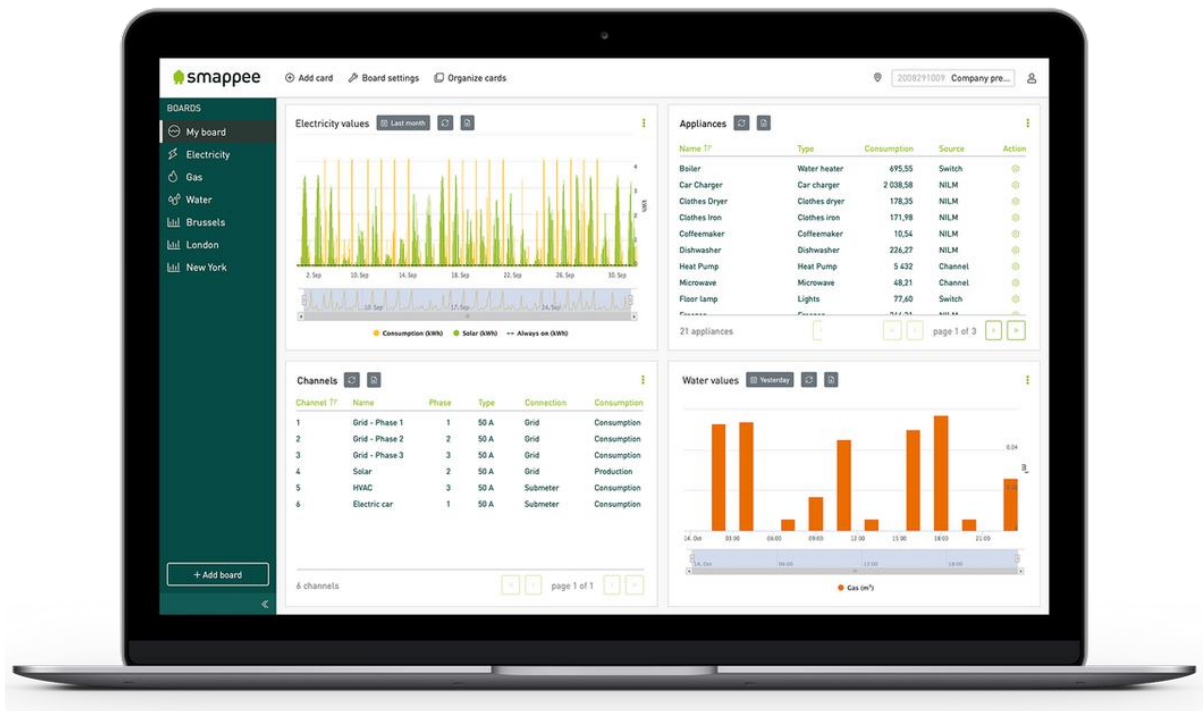


Figura 2.8. Dashboard web de Smappee.

Mediante la recolección de datos y análisis de los consumos, tiempos de uso y demás parámetros, los usuarios adquieren características de funcionamiento inherentes a sus hogares. De este modo ellos pueden realizar un control más óptimo respecto a su propio consumo consiguiendo así un ahorro de energía.

Como se ha podido presentar en los dos ejemplos reales que están disponibles en el mercado, existen diferentes soluciones que contemplan de una forma u otra, la gestión de la energía en el hogar. En términos generales, estos sistemas HEMS ofrecen las siguientes características:

- Aplicación web o móvil para presentación de datos.
- Hub de control de hardware para dispositivos inteligentes.
- Alarmas.
- Análisis de consumos de energía.
- Análisis de energía solar.

Sin embargo, de las características antes mencionadas se puede observar que las acciones a tomar sobre el uso de los dispositivos no son propuestas de forma automática, es decir, en base a los datos medidos por los sensores y recolectados en el sistema HEMS,

se identifican puntos a mejorar como daños o el consumo en horas picos, pero es decisión del usuario las acciones a tomar para poder mejorar la situación detectada.

Por otra parte, en ninguna de las aplicaciones encontradas se ha podido comprobar que tengan alguna característica que permita analizar o contrastar los consumos respecto al costo de la energía, lo cual es un punto de vital importancia para el presente proyecto donde se busca crear una herramienta que pueda optimizar los consumos de un hogar para poder reducir el costo de la energía facturada.

2.3. Investigaciones más recientes

Alrededor de los últimos 8 años, los sistemas de gestión de energía en los hogares HEMS han sido un tema muy desarrollado en la comunidad de investigadores a nivel mundial. En (Shareef et al., 2018) se presenta una visión general de las investigaciones acerca de los HEMS de donde se extrae la siguiente reseña. Una de las primeras investigaciones sobre los HEMS data del año 1979, donde se hace énfasis en los microprocesadores como los dispositivos encargados de llevar el poder de procesamiento. En otro artículo publicado en 1982, surge el primer algoritmo de optimización para la gestión de la energía enfocado en reducir la demanda y el tiempo de uso. Investigaciones posteriores han hecho énfasis en el uso de cargas específicas como luz o aire acondicionado, llegando así a las publicaciones más actuales donde se puede evidenciar que, el principal interés radica en la interacción de tecnologías móviles con algoritmos de planificación de tareas que generan cronogramas de uso óptimos basados en dos factores fundamentales; el primero es el consumo de energía en función de la variación dinámica del costo de la misma, mientras que el segundo es el impacto en la satisfacción del cliente.

Existen varias formas de buscar el cumplimiento de estos lineamientos importantes cuando se trata de crear o construir un HEMS. El primer punto de interés está relacionado con la formulación del problema, donde se han presentado diversas propuestas de las que se destaca que todas empiezan por la correcta categorización de los artefactos en el hogar. Esta clasificación considera básicamente artefactos *diferibles* (dentro de los cuales están los *flexibles* y *no flexibles*) cuyo inicio de operación puede ser movido a otro momento del día, los artefactos *reducibles* que pueden ser apagados durante su uso con la necesidad de encenderlos posteriormente para finalizar el ciclo, los artefactos *térmicos* encargados de mantener una cierta temperatura, y los artefactos críticos, que no pueden ser controlados por el sistema (Althaher, Mancarella, & Mutale, 2015). Una vez realizada esta clasificación, se continúa con el modelamiento de los electrodomésticos.

El segundo punto de interés es el algoritmo de optimización utilizado para poder generar la planificación o cronograma de operación de los electrodomésticos antes clasificados. En consideración de este aspecto, se han generado diversas publicaciones con varios métodos de los cuales destacan: el *Algoritmo de Aproximación Externa* (Althaher et al., 2015), el *Algoritmo Genético*, un algoritmo de optimización global basado en la evolución (Zhuang Zhao, Won Cheol Lee, Yoan Shin, & Kyung-Bin Song,

2013), el *Algoritmo de Optimización por Enjambre de Partículas* (PSO, por sus siglas en inglés), un algoritmo estocástico de optimización global (Huang, Tian, & Wang, 2015) y el *Algoritmo de Colonia de Abejas Artificiales* (ABC, por sus siglas en inglés), que también es un algoritmo estocástico (Zhang, Zeng, Li, Zang, & Li, 2015).

En todas estas investigaciones se han propuesto esquemas de funcionamiento donde el objetivo principal es la optimización. Cada artículo analiza los factores involucrados de distinta manera, pero casi todos los más actuales incorporan la respuesta ante la dinámica del costo de la energía, a la vez que consideran el factor de satisfacción del cliente. Al final, el resultado es la solución óptima global que cumple con todas las limitaciones del modelamiento del problema.

3. ANÁLISIS Y SELECCIÓN DE HERRAMIENTAS

En este apartado se analizan las diferentes herramientas necesarias para el desarrollo del presente proyecto. Comprende puntos importantes como el sistema operativo de funcionamiento, la plataforma sobre la cual se desarrollará la aplicación, entre otras. Además, se analiza el algoritmo de optimización más adecuado de entre los mencionados en el apartado 2.3, que será usado como parte de la programación. Finalmente, en base a criterios técnicos y objetivos, se seleccionan todas las partes del entorno de trabajo requerido, que serán implementadas a lo largo de todo el desarrollo.

3.1. Sistema operativo.

Existen principalmente tres sistemas operativos muy utilizados a nivel mundial donde cada uno tiene sus características y prestaciones específicas.

3.1.1. Windows

Windows es un sistema operativo propietario, creado por la empresa *Microsoft Corporation*[®]. Este está diseñado para funcionar principalmente en computadoras personales basados en arquitectura Intel. Windows es el sistema operativo más utilizado a nivel mundial, con alrededor del 80% de computadoras conectadas a internet (StatCounter, s. f.).

Debido a su popularidad, casi todo software que se ofrece en el mercado es diseñado para este sistema operativo o tiene una versión compatible, con lo cual se puede tener disponible cualquier herramienta que sea necesaria funcionando sin problema. Sin embargo, una característica negativa es la amplia difusión de software malicioso o virus, que aprovechan las vulnerabilidades del sistema para realizar acciones no permitidas o peligrosas para los usuarios.

La versión más actual es *Windows 10*, que ofrece una interfaz gráfica muy amigable. El costo de la versión *Windows 10 Home* según la página oficial de Windows es de 145€ (Corporation, s. f.)

3.1.2. Linux

El sistema operativo Linux es uno de los sistemas más populares en el ámbito de la informática. Sin embargo, según (StatCounter, s. f.), actualmente nada más alcanza una participación inferior al 2% en el mercado de las computadoras; aunque por otra parte, este sistema operativo ha sido la base para muchos sistemas móviles tales como Android o Chrome OS, incluso ofreciendo versiones para *IoT*.

Una de sus grandes ventajas es ser ampliamente configurable. Al ser un software libre, todo su código de desarrollo está disponible para su estudio y mejoras, lo que permite conocer a detalle sus características y donde se puede realizar configuraciones específicas según sea necesario, sin embargo, necesita un nivel importante de conocimientos informáticos. En consecuencia, Linux es el sistema más utilizado en supercomputadoras y a nivel de servidores (IDC, 2012), donde los usuarios tienen el nivel

técnico suficiente para manipular el sistema operativo. Una de las versiones más conocidas de este sistema es *Ubuntu*, que actualmente está en la versión 18.10

Respecto al costo de la licencia de uso, ya que Linux está categorizado como software libre, puede ser instalado sin necesidad de realizar ningún pago.

3.1.3. macOS

macOS es el sistema operativo de la marca *Apple*[®]. Este sistema operativo es muy popular en el área de edición de videos o imágenes por su potencia de procesamiento y facilidad de manejo. Según (StatCounter, s. f.), actualmente cuenta con un 13% de participación en el mercado de las computadoras.

Una limitación grande es que, al ser un software propietario, no puede ser utilizado en cualquier computador. De hecho, macOS es un sistema creado exclusivamente para las computadoras vendidas por la misma marca. Actualmente se está distribuyendo la versión más reciente denominada macOS *Mojave*.

En cuanto a seguridad, macOS se ha caracterizado por ser un sistema operativo muy seguro. Al ser un software propietario que nada más funciona en dispositivos de la misma empresa, la corrección o aparición de vulnerabilidades se ha mantenido bastante controlado mientras que la creación de software malicioso se ha sido limitada, lo cual supone una gran ventaja para los usuarios.

A pesar de no ser un software libre, macOS es un sistema operativo que se distribuye de forma gratuita en la actualidad, considerando que solo usuarios de computadoras de *Apple*[®] pueden acceder a este software (Apple, s. f.).

3.2. Entorno de desarrollo.

En lo que respecta a entornos de desarrollo o IDE (*Integrated Development Environment*), es un conjunto de herramientas digitales que permitirán el desarrollo de la aplicación HEMS. Como puntos principales se deben diferenciar dos componentes importantes respecto al entorno. El primero es el desarrollo de la interfaz gráfica (conocido como *front-end*), mientras que el segundo es la programación y configuración de herramientas que trabajan por detrás de la aplicación como por ejemplo la base de datos o los algoritmos de seguridad (conocido como *back - end*).

3.2.1. Laravel

Laravel es un entorno de desarrollo web, o *framework*, basado en el lenguaje de programación PHP. Este entorno de trabajo es de código abierto, lo cual indica que es gratis y libre para su uso. Está pensado para el desarrollo de aplicaciones web siguiendo el patrón MVC (*model - view - controller*). La versión actual de Laravel es la 5.8

En cuanto a sus características importantes respecto al *back - end*, mediante comandos se puede generar la programación sobre control de acceso de usuarios de forma muy rápida. También presenta varias maneras de acceder a bases de datos relacionales de

una manera relativamente sencilla gracias a su ORM (*Object-Relational Mapping*) *Eloquent*. Asimismo, dentro de su estructura básica, permite el manejo de eventos, ante los cuales se pueden disparar varias acciones programadas. Además, se puede hacer uso de sus herramientas de notificación para configurar las alarmas necesarias (Laravel, s. f.-a).

Respecto al *front-end*, Laravel ofrece dentro de sus características básicas unas plantillas de desarrollo donde se pueden generar *layouts* que simplifican el desarrollo de la interfaz de usuario y también su experiencia de uso. Por otra parte, Laravel se puede integrar con otros *frameworks* especializados en *front-end*, destacándose entre varios *Vue.js*, que permite la creación de componentes reutilizables e interactivos como botones, gráficas temporales, entre otras, con los cuales se construye la interfaz visual de la aplicación (*Vue.js*, s. f.).

El uso de un *framework* web como Laravel supone un grado de complejidad mayor ya que, si bien es cierto existen muchos ejemplos en repositorios de código abierto, se requiere un conocimiento más especializado para poder desarrollar la programación hasta consolidar una aplicación funcional.

3.2.2. LabVIEW

LabVIEW (acrónimo de *Laboratory Virtual Instrument Engineering Workbench*) es una herramienta digital ampliamente utilizado en el campo de la ingeniería. Destaca principalmente por su programación gráfica (o lenguaje G), la cual se desarrolla mediante diagrama de bloques, permitiendo una interacción más intuitiva con los usuarios. Además, este software permite visualizar la aplicación en cada etapa del desarrollo (Travis & Kring, 2007). La última versión lanzada al mercado es LabVIEW 2018, anunciada en noviembre de 2018.

Durante el uso de este programa se crean instrumentos virtuales, donde se definen dos ventanas de trabajo, una para la parte de la interfaz gráfica (*front-end*), y otra para la programación del funcionamiento de la aplicación (*back-end*). Dentro de sus ventajas, LabVIEW ofrece gran variedad de librerías ya desarrolladas para diferentes aspectos como adquisición de datos, generación de señales, matemáticas, estadística, entre otras. También se destaca la capacidad de conexión con dispositivos externos como microprocesadores, cámaras, sensores, etc. (National Instruments, s. f.-a). Sin embargo, LabVIEW muestra ciertas desventajas como el hecho de que por más pequeña que sea la aplicación desarrollada, se debe correr el ambiente de funcionamiento completo lo cual hace un poco lento el proceso. También, al ser un lenguaje de programación gráfico, se pierden características como control de versiones y las comparaciones de código textual (National Instruments, s. f.-c).

LabVIEW es un software propietario, perteneciente a la empresa *National Instruments* por lo que su uso está condicionado al pago de la licencia. El costo de su versión *LabVIEW Completo* es de 3451.00€, mientras que la versión *LabVIEW Profesional* cuesta 5747.00€ (National Instruments, s. f.-b)

3.2.3. Matlab

MATLAB (*matrix laboratory*), software desarrollado por la empresa *MathWorks*[®], es un ambiente de computación multiparadigma. Esta herramienta es muy popular en el ámbito de la ingeniería e investigación, y en especial en el mundo académico. Está basado en un lenguaje de programación tipo *script* (MathWorks, s. f.-a). Dentro de sus características principales están la manipulación numérica de matrices, implementación de algoritmos y la creación de interfaces gráficas. En su instalación básica, MATLAB ofrece muchas herramientas o *toolbox* para varias tareas, de las cuales resulta relevante el paquete de optimización.

Otro aspecto muy interesante acerca de esta plataforma de trabajo es que consta de una amplia lista de librerías y subprogramas con los cuales se puede comunicar e interactuar, incluso siendo estos escritos en otros lenguajes de programación como C, C++, Python, entre otros.

De acuerdo a *MathWorks*, en el 2018, MATLAB contaba con más de 3 millones de usuarios en todo el mundo (MathWorks, 2018).

Al ser un software propietario, el uso de este programa requiere la compra del programa o de su licencia de uso. Según la página web oficial de la empresa, existen varias categorías bajo la cual se venden las licencias como standard, personal, educación y estudiante. Además del costo del programa principal MATLAB, existen costos asociados a las diferentes herramientas que se pueden llegar a necesitar en el desarrollo de las aplicaciones. Para una licencia perpetua del programa en la categoría standard, se requiere el pago de 2000.00€ (MathWorks, s. f.-b)

3.2.4. Visual Studio

Visual Studio es un entorno de desarrollo integrado (IDE) de la compañía Microsoft. Ofrece una amplia variedad de lenguajes de programación y es muy utilizado para el desarrollo de software para computadores, servicios web y también aplicaciones móviles. En esta se puede desarrollar tanto la parte funcional del programa como por ejemplo formularios, bases de datos, entre otros, así como también la interfaz gráfica de usuario (UI).

Dentro de sus principales herramientas, VS cuenta con un editor de texto muy potente el cual cuenta con resaltado de sintaxis y auto completado utilizando *IntelliSense* para variables, funciones, métodos y bucles. También, VS ofrece una herramienta para depuración de código o *debugger* para cualquier lenguaje de programación soportado, el cual incorpora el uso de *breakpoints* y *watches*, que son de vital importancia durante el desarrollo de un programa (MSDN, s. f.-b). En lo que respecta al diseño, VS incluye herramientas de soporte como el Diseñador de Formularios de Windows (WFD) que permite la creación rápida y de forma visual de formularios para la interacción de datos ingresados por el usuario y bases de datos o consultas (MSDN, s. f.-a).

Existen tres tipos de licencia bajo la cual se puede utilizar este software. La versión Comunidad es distribuido de forma gratuita para usos particulares y colaboración en

proyectos de código abierto; la versión Profesional con el costo por el primer año de 1199.00€ (799€ por año de renovación), es adecuado para equipos de trabajo pequeños mientras que la versión Empresa ofertada a 5999.00€ por el primer año (2569€ por año de renovación), es aplicable para equipos de trabajo medianos o grandes (Microsoft, s. f.).

3.3. Algoritmo de optimización.

Existen varios métodos mediante los cuales se puede buscar una solución óptima, la cual depende directamente del tipo de problema que se tiene y el enfoque que se quiere adoptar. Uno de los algoritmos más investigado para tareas de planificación es el algoritmo genético, que pertenece al grupo de algoritmos evolutivos. Otro ejemplo es el algoritmo de optimización por enjambre de partículas, también conocido como PSO. Considerando las diferentes opciones encontradas en la investigación, se consideran las siguientes opciones como principales.

3.3.1. Algoritmo Genético

El algoritmo genético (AG) es una técnica de optimización y búsqueda basada en los principios de la genética y la selección natural. Esta técnica fue descrita inicialmente por John Holland (1975); quien lo presentó como un algoritmo que permite a una población de individuos evolucionar hacia un estado que maximice la “bondad” de cada solución, es decir, minimice la función de costo (Sivanandam & Deppa, 2008)

El algoritmo genético está compuesto por varios factores relacionados con los principios de la genética, donde cada uno tiene una función específica para llegar a la solución del problema.

La **Población** representa el conjunto de cromosomas y es la unidad que usa el algoritmo genético para encontrar la solución más óptima. Esta población inicial sirve de base para las futuras generaciones (Haupt & Haupt, 2004).

Los **Cromosomas** representan un conjunto de genes y contiene la información de todo un arreglo. Para efectos prácticos, cada cromosoma contiene la información de una potencial solución para el problema planteado.

Finalmente, el **Gen** es el componente más pequeño dentro del algoritmo, el cual porta la información de cada parte del cromosoma, es decir, contiene las partes de la potencial solución.

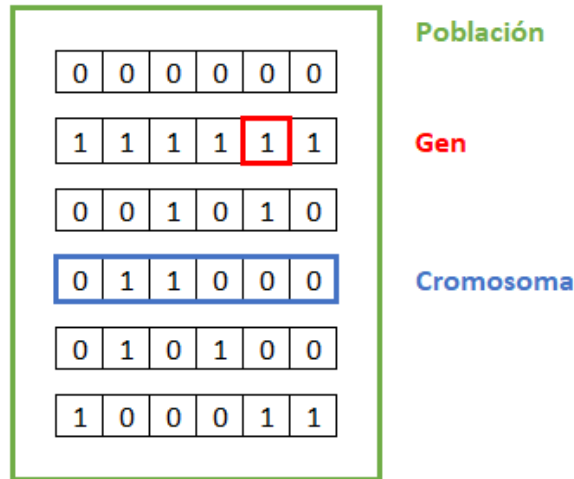


Figura 3.1. Partes del algoritmo genético.

Para determinar el costo de cada potencial solución, se debe crear una función de evaluación (o *fitness function*) que considere las limitaciones del problema.

Otro aspecto muy importante del algoritmo genético es el método mediante el cual se generan nuevas poblaciones para la búsqueda de la solución más óptima. Dentro de las operaciones requeridas están la selección, el cruce y la mutación (Haupt & Haupt, 2004).

La operación de **selección** consiste en determinar los individuos o cromosomas mejor puntuados para posteriormente operar con ellos para obtener nuevos individuos. A estos individuos se los denomina padres. La operación de **cruce** por otra parte, como su nombre lo indica, consiste en cruzar los padres previamente seleccionados, y así obtener nuevos individuos (hijos) dentro de la población. Finalmente, el proceso de **mutación** consiste en modificar uno o varios genes de un individuo para poder explorar nuevas posibles soluciones.

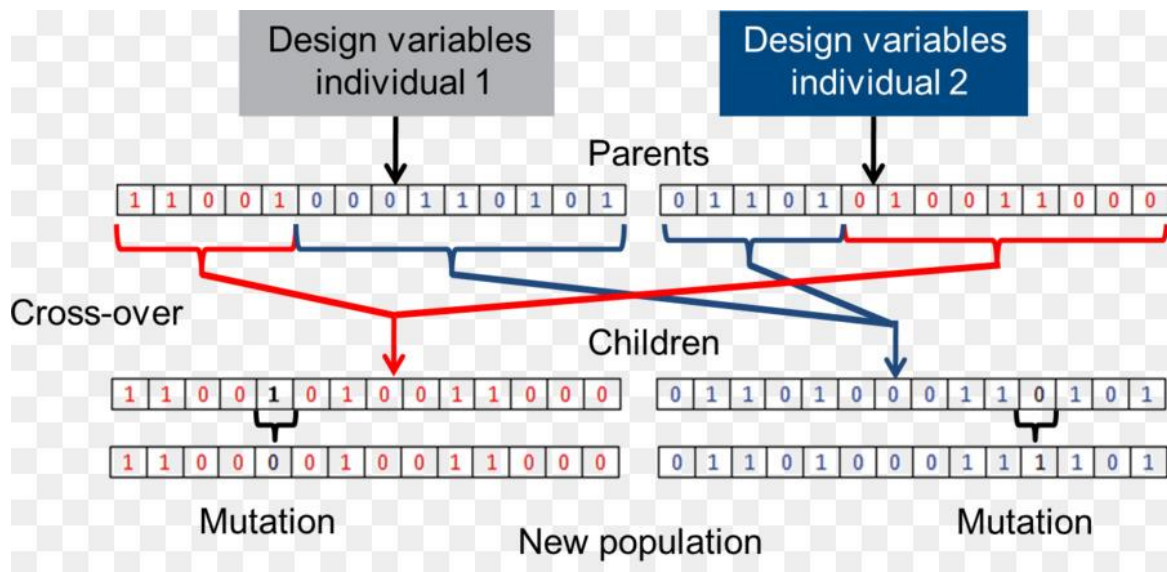


Figura 3.2. Operaciones del algoritmo genético.

Es importante destacar que existen varias formas de llevar a cabo cada operación y que depende del diseño del problema. Asimismo, se debe indicar que la operación de mutación debe ser aplicada de forma cuidadosa ya que, si bien es cierto ayuda a salir del campo de una solución local, también puede generar la salida del campo de la solución global sin ser necesario.

El algoritmo genético completo sigue el diagrama de flujo de la Figura 3.3

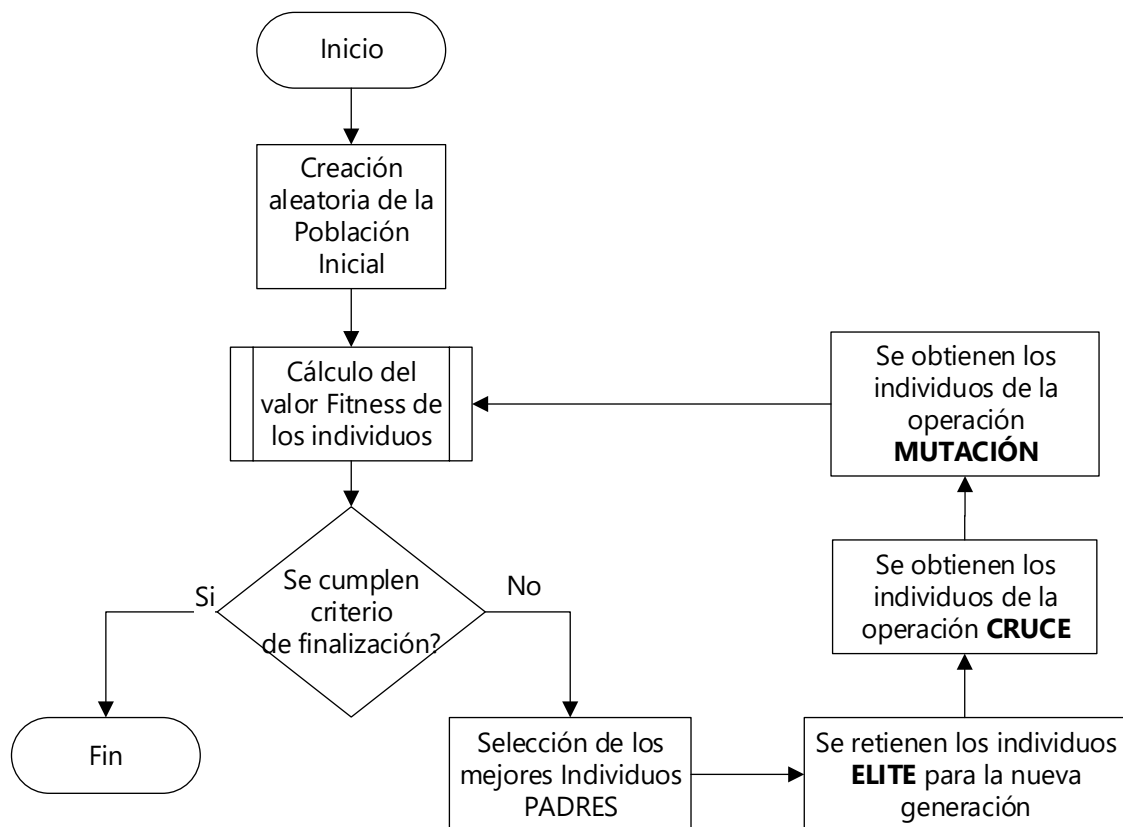


Figura 3.3. Diagrama de flujo del algoritmo genético.

3.3.2. Optimización por enjambre de partículas

El algoritmo PSO o de optimización por enjambre de partículas es una técnica categorizada como “foraging” social. Esto quiere decir que se busca simular las conductas orientadas a conseguir alimento de varias especies como las abejas, hormigas, entre otras.

El PSO es un algoritmo metaheurístico, es decir que requiere poca información sobre el problema, mientras que puede buscar una cantidad grande de espacios de soluciones. Es un algoritmo iterativo donde cada partícula explora un espacio específico, pero no se puede garantizar que la solución sea la más óptima (Espitia Cüchango & Sofrony Esmeral, 2015).

Este algoritmo se basa en un conjunto de individuos que realizan una búsqueda en grupo, en un entorno determinado para detectar alimentos, evitar obstáculos o guiar al grupo de manera colectiva a mejores ubicaciones (Torres & Baran, 2015).

El enjambre describe un comportamiento específico, donde cada partícula tiene memoria y puede retener información de su estado anterior. Las partículas se mueven de forma circular sobre un punto denominado vórtice. Cada partícula se mueve influenciada por dos parámetros, la individualidad y la sociabilidad. El primero se define como la tendencia a volver a situarse a la mejor posición previa, mientras que la sociabilidad es la tendencia de moverse a la mejor posición previa del enjambre entero (Espitia Cüchango & Sofrony Esmeral, 2015).

Al igual que el primer algoritmo descrito, el PSO consiste en encontrar el valor mínimo de una función objetivo, el cual no es necesario que el enjambre lo conozca. Existen dos tipos de algoritmos básicos para determinar la posición y velocidad de las partículas; el primero es el algoritmo binario donde la posición de la partícula se determina mediante una distribución de probabilidad (Torres & Baran, 2015). El segundo es el algoritmo continuo donde la partícula es expresada con sus valores reales dentro del espacio de búsqueda.

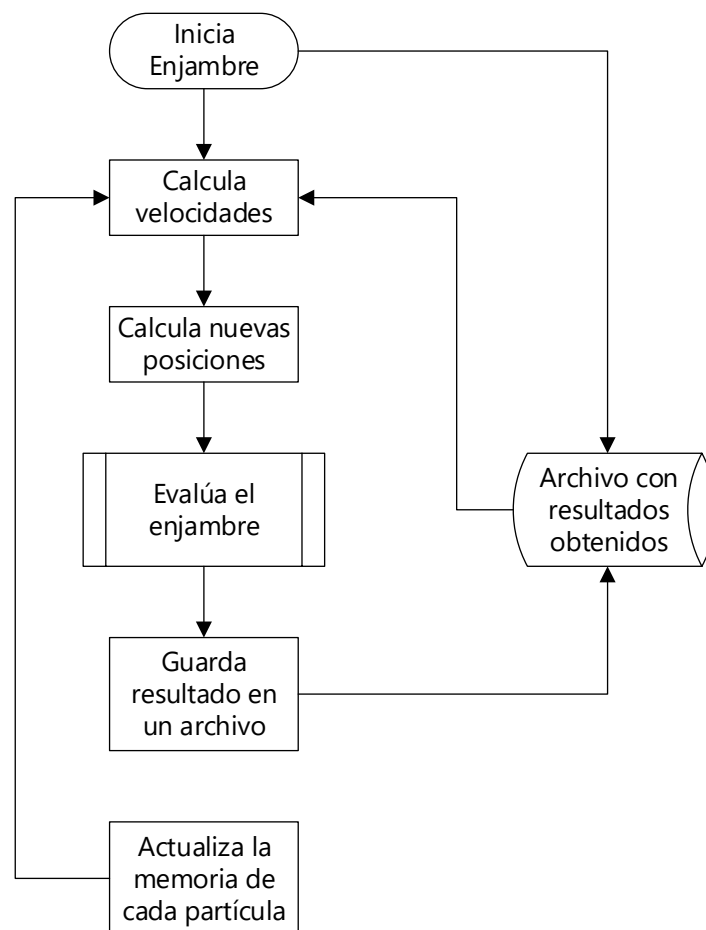


Figura 3.4. Diagrama de flujo del algoritmo PSO.

3.3.3. Colonia Artificial de Abejas

La Colonia Artificial de Abejas, conocido como ABC por sus siglas en inglés (*Artificial Bee Colony*) es una técnica de optimización que simula el comportamiento de una colonia de abejas. El ABC pertenece al grupo de algoritmos de inteligencia de enjambre, donde se cumplen dos características fundamentales; una es la auto organización del sistema total mediante la interacción de los componentes de bajo nivel y otra es la división del trabajo donde diferentes individuos desempeñan tareas de manera simultánea (Karaboga, 2005).

En el modelo del ABC existen tres componentes. El primero es la fuente de alimentación, valorado por la proximidad al panal, la riqueza y concentración de su energía y la facilidad de extraer esa energía. El segundo componente son las abejas empleadas, que están asociadas a una sola fuente de alimentación que es donde se encuentran actualmente trabajando, y portan toda la información relevante de esta fuente. El tercer componente son las abejas no empleadas que se encuentran constantemente en la búsqueda de nuevas fuentes de alimentación para explotar y son de dos tipos; las exploradoras están constantemente buscando fuentes de alimentación en los alrededores del panal, mientras que las espectadoras se mantienen en el panal y reúnen toda la información de las demás abejas acerca de las fuentes (Karaboga, 2005).

En el ABC, cada fuente de alimentación cuenta solamente con una abeja empleada. La información es comunicada a las abejas espectadoras mediante una danza. Una vez que el néctar de esta fuente ha sido agotada, esta abeja se convierte en una exploradora y empieza una búsqueda aleatoria de una nueva fuente de alimentación (Zhang et al., 2015).

El algoritmo ABC puede ser descrito mediante la secuencia de pasos descrita en la Figura 3.5 (Qiu, Shen, Xie, & Wang, 2013).

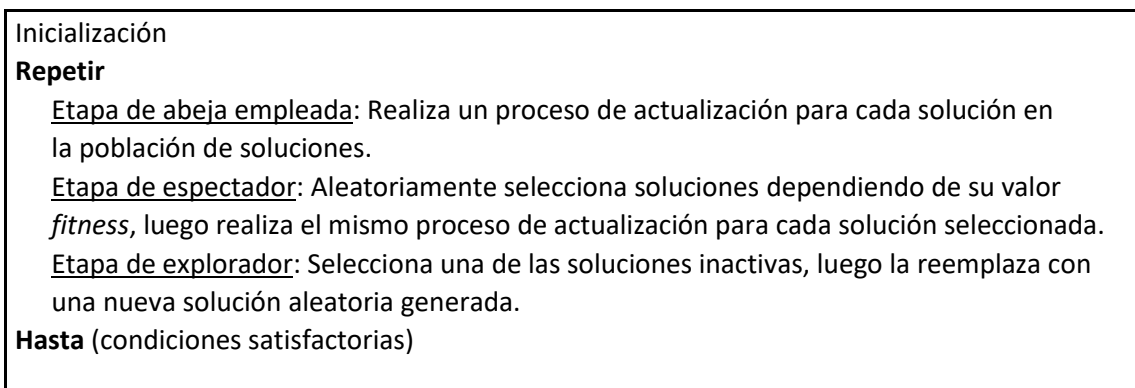


Figura 3.5. Algoritmo del método ABC.

3.4. Selección de herramientas

En esta sección se presenta la selección de las herramientas previamente descritas. Para este punto se ha de determinar un método objetivo para evaluar cada opción.

3.4.1. Criterios de selección.

Para poder determinar las herramientas más idóneas para la implementación del presente proyecto, se ha elegido el método del valor técnico ponderado VTP. En este método es necesario establecer varios criterios para cada herramienta y definir el **peso** de cada uno de ellos según la importancia y necesidad para el presente desarrollo. Luego, se ha de puntuar cada herramienta bajo cada uno de los criterios establecidos lo cual entregará una puntuación final, donde la mayor nos indicará la herramienta más apta.

Respecto al **sistema operativo**, se han identificado los siguientes criterios: costo de adquisición, compatibilidad con programas de desarrollo, seguridad, estabilidad de funcionamiento y soporte técnico por parte de los desarrolladores.

Tabla 3.1. Criterios de selección del sistema operativo.

Ítem	Criterio de selección	Relación de interés	Peso
1	Costo	MENOR	5
2	Compatibilidad	MAYOR	4
3	Seguridad	MAYOR	4
4	Estabilidad	MAYOR	3
5	Soporte técnico	MAYOR	3

En cuanto al **entorno de desarrollo**, los criterios identificados son el costo, la compatibilidad en las diferentes plataformas, estabilidad, soporte por parte de las comunidades de desarrollo que utilizan la plataforma, también la complejidad y la capacitación que se requiere para poder utilizarlos.

Tabla 3.2. Criterios de selección del entorno de desarrollo.

Ítem	Criterio de selección	Relación de interés	Peso
1	Costo	MENOR	5
2	Compatibilidad	MAYOR	4
3	Estabilidad	MAYOR	3
4	Soporte (comunidades online)	MAYOR	3
5	Capacitación requerida	MENOR	5
6	Complejidad	MENOR	3

Finalmente, los criterios identificados para la selección del **algoritmo de optimización** son el conocimiento específico que se requiere sobre este, la facilidad de implementación, la efectividad frente a problemas de planificación o generación de cronogramas, el costo computacional y las fuentes de investigación disponibles tales como publicaciones científicas, proyectos ejemplo que utilicen el método, entre otros.

Tabla 3.3. Criterios de selección del algoritmo de optimización.

Ítem	Criterio de selección	Relación de interés	Peso
1	Conocimiento específico	MENOR	4
2	Facilidad de implementación	MAYOR	4
3	Efectividad	MAYOR	4
4	Costo computacional	MAYOR	5
5	Fuentes de investigación	MAYOR	3

3.4.2. Herramientas seleccionadas.

Después de haber realizado los cálculos del valor técnico ponderado (ver el ANEXO A – Valor Técnico Ponderado), se han seleccionado **Linux** como el sistema operativo de desarrollo, **Laravel** como la plataforma de desarrollo donde se construirá la aplicación basada en la web, sobre la cual se programará el **Algoritmo Genético** como algoritmo de optimización.

Tabla 3.4. VTP para el sistema operativo.

Criterio de selección	Sistema Operativo			Peso
	Windows	Linux	macOS	
Costo	0	50	50	50
Compatibilidad	36	28	16	40
Seguridad	24	32	32	40
Estabilidad	15	27	24	30
Soporte	21	27	21	30
VTP	0,51	0,86	0,75	190

Tabla 3.5. VTP para el entorno de desarrollo.

Criterio de selección	Software de desarrollo				Peso
	Laravel	LabVIEW	MatLAB	Visual Studio	
Costo	50	25	25	50	50
Compatibilidad	40	28	40	36	40
Estabilidad	24	24	27	21	30
Soporte	36	28	28	28	40
Capacitación requerida	25	30	25	30	50
Complejidad de uso	18	24	24	21	30
VTP	0,80	0,66	0,70	0,78	230

Tabla 3.6. VTP para el algoritmo de optimización.

Criterio de selección	Algoritmo de Optimización			Peso
	GA	PSO	ABC	
Conocimiento específico	28	28	28	40
Facilidad de implementación	45	35	35	50
Efectividad	40	32	32	40
Costo computacional	40	40	35	50
Fuentes de investigación	24	18	15	30
VTP	0,84	0,73	0,69	210

4. DESARROLLO DE LA APLICACIÓN HEMS

En esta sección se presenta el desarrollo de las partes que componen el proyecto. El desarrollo comprende la sección de diseño del sistema de generación fotovoltaica, el análisis del costo dinámico de la energía eléctrica, el modelamiento matemático del sistema que se quiere controlar y finalmente la programación de la aplicación para obtener un producto web.

4.1. Sistema de generación fotovoltaica.

Una variable importante para el desarrollo del algoritmo de optimización es la energía fotovoltaica que se produce en el día. Para ello, se modelará una instalación de generación mediante la herramienta SAM.

El *System Advisor Model* (SAM) es un software diseñado para facilitar la toma de decisiones en nuevos proyectos que incluyen energías renovables. Una de las opciones más utilizadas es el modelamiento de sistemas de generación fotovoltaica en el techo de hogares (NREL, s. f.). En este aspecto, SAM cuenta con una base de datos de elementos como paneles solares, baterías, onduladores, entre otros, para realizar un correcto dimensionamiento y cálculos técnico-económicos durante la fase de planificación y diseño de un proyecto nuevo.

4.1.1. Condiciones de instalación

Para poder modelar correctamente una instalación de generación de energía fotovoltaica, en primer lugar, se deben estudiar y definir las condiciones físicas y requerimientos locales de la instalación. Como punto importante se debe indicar que la casa sobre la cual se realiza el proyecto, se encuentra ubicada en Ecuador. El sistema de generación que se diseñará, es en base al área de instalación mostrada en la Figura 4.1.



Figura 4.1. Tejado disponible para la instalación de los paneles de generación PV.

En la Tabla 4.1 se resume todas las características principales para poder realizar la actividad de diseño del sistema de generación fotovoltaica.

Tabla 4.1. Condiciones para la instalación de la generación fotovoltaica.

Condiciones	Valor
Ubicación	
Latitud	$-1,83^{\circ}N$
Longitud	$-78,18^{\circ}E$
Inclinación	10°
Dimensiones	
Largo	8 m
Ancho	4 m
Superficie	$32 m^2$
Eléctrico	
Voltaje	120 V_{AC}
Tipo de instalación	Conectada a red

4.1.2. Diseño de la instalación

Una vez definidas las condiciones de instalación, se debe seleccionar el tipo de panel solar que se instalará. Existen varias opciones de venta en el mercado cuyas características se encuentran detalladas dentro de la herramienta SAM para la realización de la simulación. Es importante escoger el tipo de tecnología del panel, sus características eléctricas como potencia máxima, voltaje, corriente, entre otras.

Para este caso en particular se ha seleccionado el panel solar SUN POWER SPR-X21-335-BLK (ver Figura 4.2) que ofrece las características que se indican en la Tabla 4.2. Estos datos e imágenes han sido obtenidos de la web del fabricante.

Tabla 4.2. Características del panel solar SUNPOWER SPR-X21-335-BLK

Característica	Valor
Marca	SUN POWER
Modelo	SPR-X21-335-BLK
Tecnología	Mono-c-Si
Clase	Class A+
Área	$1,63 m^2$
Largo	1,558 m
Ancho	1,046 m
Eficiencia Nominal	20,56%
Potencia máxima pico	335,21 W_{DC}
Voltaje de máxima potencia	57,3 V_{DC}
Corriente de máxima potencia	5,8 A_{DC}
Voltaje de circuito abierto	67,9 V_{DC}
Corriente de circuito abierto	6,2 A_{DC}



SIGNATURE™ BLACK
SPR-X21-335-BLK

Figura 4.2. Panel solar SPR-X21-335-BLK de la marca Sun Power

Considerando el área total disponible para la instalación, y las dimensiones del panel seleccionado, se ha diseñado el *layout* que se muestra en Figura 4.3. En esta configuración se cuenta con dos ramas paralelas, donde cada una contiene seis paneles solares conectados en serie

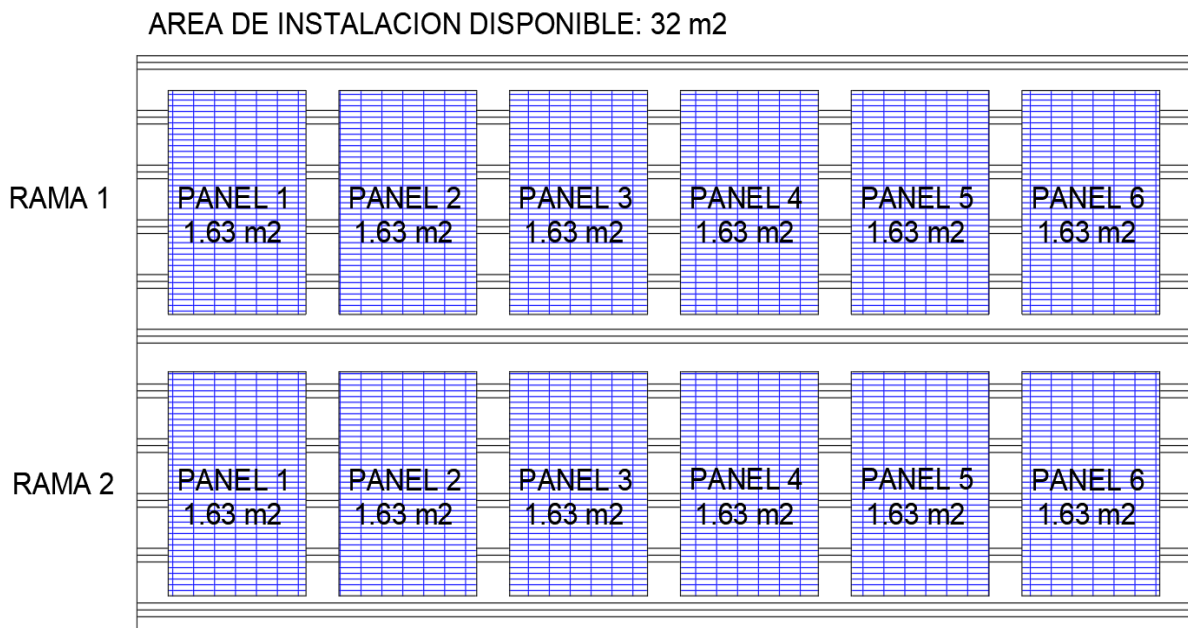


Figura 4.3. Layout de la instalación de generación fotovoltaica.

Tomando en cuenta que cada panel entrega las siguientes características eléctricas

$$P_p = 335,21 W_{DC}$$
$$V_{mp} = 57.3 V_{DC}$$
$$I_{mp} = 5.8 A_{DC}$$

Se puede obtener como total del sistema

$$V_{total} = 57.3 * 6 = 343.8 V_{DC}$$

$$I_{total} = 5.8 * 2 = 11.6 A_{DC}$$

$$P_{total} = 3988 W_{DC}$$

Después que se han calculado los valores de voltaje, corriente y potencia máxima que se puede obtener del sistema, se procede a seleccionar el inversor que permita el manejo de dichas variables.

El inversor *Power Station PS247-05-180* ofrece las características que se muestran en la Tabla 4.3

Tabla 4.3. Características técnicas del inversor POWER STATION PS247-05-180

Característica	Valor
Marca	POWER STATION
Modelo	PS247-05-180
Potencia máxima AC	5020 W_{AC}
Potencia máxima DC	5414,17 W_{AC}
Voltaje nominal	120 V_{AC}
Voltaje máximo	420 V_{DC}
Corriente máxima	17,47 A_{DC}



Figura 4.4. Inversor PS247-05-180

El sistema que se está modelando no contará con un sistema de seguimiento de sol, es decir, la instalación de los paneles será fija. Además, los paneles estarán instalados con una inclinación de 10° con orientación hacia el norte (azimuth = 0°).

4.1.3. Simulación de la instalación

Después de haber diseñado las partes principales de la instalación, se continua con el proceso de simulación mediante la herramienta SAM. Dicha herramienta servirá para crear la base de datos de generación de energía eléctrica fotovoltaica.

Además de los datos de diseño, SAM requiere de información adicional como sombras, pérdidas en los componentes, baterías y costos, entre otros. Sin embargo, ya que en este caso solo interesa conocer la parte de generación eléctrica, se dejarán los demás valores por defecto. Cabe mencionar que se ha utilizado la base de datos de la NASA para obtener las variables climáticas de la localización como temperatura, irradiancia, etc.

Después de correr el simulador de SAM, se ha obtenido una extensa lista de variables relacionadas con la instalación. En la Figura 4.5 se muestra, a manera de ejemplo, el perfil de energía generada para tres días aleatorios obtenidos después de la simulación.

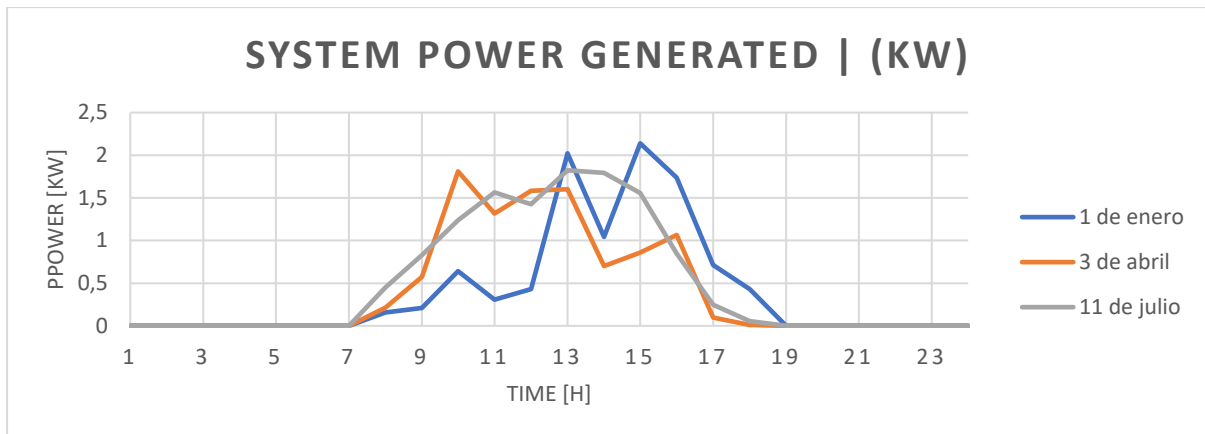


Figura 4.5. Perfil de generación fotovoltaica estimada, para 3 días específicos.

En la Figura 4.6 se muestra el perfil de energía generada del día característico de cada mes de un año meteorológico típico.

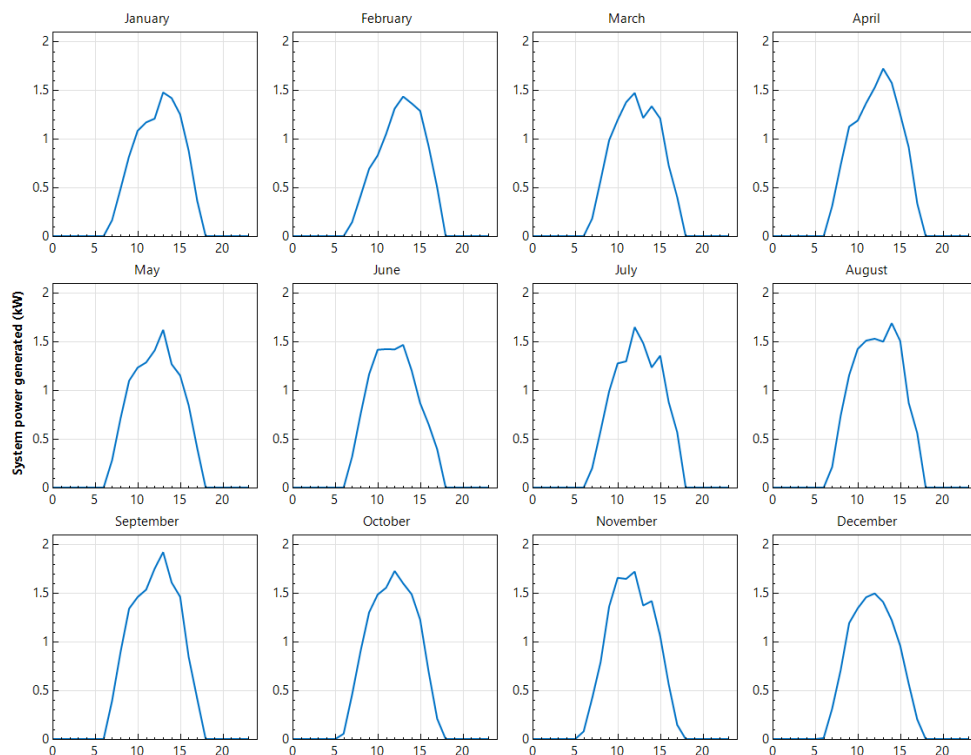


Figura 4.6. Perfiles de potencia generada del día característico de cada mes, de un año meteorológico típico.

La base de datos que se ha generado mediante la herramienta SAM, contiene el valor de energía eléctrica generada expresada en vatios, para cada hora del día, para los 365 días de año. La información es exportada en un fichero con formato CSV para su posterior uso.

4.2. Costo dinámico de la energía eléctrica.

Una de las variables de mayor interés durante el desarrollo de la aplicación HEMS es el costo de la energía. Si bien es cierto, la energía en el Ecuador es cobrada a sus usuarios mediante una tarifa fija, dependiendo el sector al que sea dirigido como por ejemplo comercial, industrial o residencial, es de mayor interés generar una herramienta capaz de trabajar con un costo de energía variable y que pueda funcionar de forma adecuada sin importar dichas variaciones. Es por eso que se ha decidido generar una base de datos con el costo de la energía tomando información de la red eléctrica de España. De este modo, la aplicación podrá alimentarse de datos que son variables a cada hora y, sobre todo, perfiles de costo reales.

En España existen 3 tarifas disponibles. La primera es la 2.0A que es la más habitual de los suministros domésticos y cuenta solo con un periodo de facturación para todo el día. La segunda es la tarifa 2.0 DHA que tiene discriminación horaria, esto quiere decir que existen dos periodos de facturación, uno para el periodo punta, y otro para el periodo valle. La tercera tarifa es la 2.0DHS, el cual difiere de la segunda tarifa en que tiene tres periodos de facturación que son el periodo punta, el periodo valle y el periodo supervalle (Selectra, s. f.).

Para construir la base de datos del costo variable, se ha tomado como referencia la tarifa 2.0 DHA (Red Eléctrica de España, s. f.). La Figura 4.7 muestra el perfil de costo para tres días aleatorios.

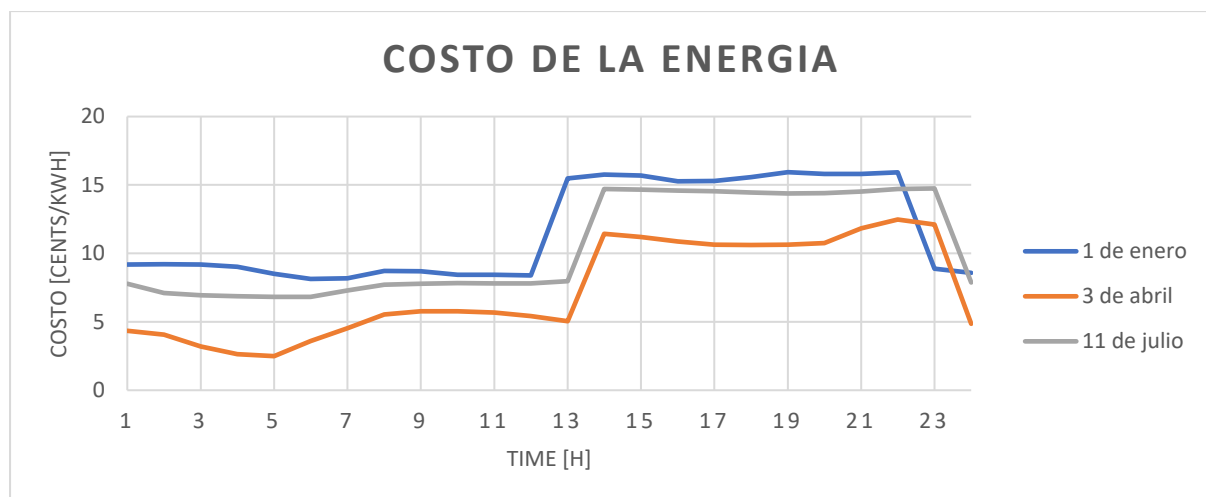


Figura 4.7. Costo horario de la energía. Tarifa 2.0 DHA

Se ha generado una base de datos con el costo de la energía horaria de los 365 días del año mediante un fichero de formato CSV.

4.3. Modelamiento del sistema.

En este apartado se presenta la formulación matemática necesaria para representar las cargas del hogar de forma adecuada.

4.3.1. Definiciones iniciales

Como punto de partida, se ha definido el periodo mínimo de computo o *timeslot* en 12 minutos, es decir, cada hora del día representa 5 *timeslots*, siendo 120 el total del día completo y se los denota con el símbolo $u \in U \triangleq \{1, 2, 3, \dots, 120\}$. Con esta resolución se puede manejar de mejor manera las cargas, además que facilita la operación del algoritmo genético (Zhuang Zhao et al., 2013).

Para poder formular el comportamiento de los artefactos operados automáticamente (AOA, de sus siglas en inglés), es importante definir ciertas variables que serán útiles para su desarrollo. Cabe mencionar que para la operación de la aplicación HEMS solo se consideraran artefactos automáticos, ya que los artefactos operados manualmente no son controlables.

A es el conjunto que contiene todos los AOA. Para cada $a \in A$ se tiene que P_a es el vector de potencia consumida planificada.

$$P_a^{(u)} \triangleq [p_a^{(1)}, p_a^{(2)}, \dots, p_a^{(120)}]$$

Donde $p_a^{(u)}$ representa el consumo de potencia del AOA a durante el *timeslot* u , de modo que es necesario definir la potencia de cada artefacto para cada uno de los 120 *timeslots*. Esto es

$$p_a^{(u)} = \frac{x_a}{5}$$

Donde x_a es el dato de potencia de cada AOA que se encuentra en placa (Zhuang Zhao et al., 2013).

Uno de los problemas que se puede generar en una tarea de planificación es que, con la finalidad de evitar los periodos donde el costo de la energía es mayor, se muevan todas las cargas a otros periodos donde dicho costo sea menor, creando un nuevo periodo de excesivo consumo que podría elevar el costo de la energía nuevamente o a su vez, sobrepasar el límite de consumo instantáneo que dispararía los disyuntores eléctricos. Es por eso es que, para el funcionamiento del algoritmo genético, se definirá un recargo al costo de la energía para estos casos. Así, el costo de la energía para cada hora, prc_h , estará definido mediante la siguiente ecuación:

$$prc_h(s_h) = \begin{cases} a_h, & \text{si } 0 \leq s_h \leq c_h \\ b_h, & \text{si } s_h > c_h \end{cases}$$

Donde s_h es la potencia total consumida durante la hora h , a_h es el costo de la energía durante la hora h en el día, b_h es el nuevo precio de la energía en caso que el consumo supere el límite c_h (Zhuang Zhao et al., 2013).

Para tener concordancia con los datos de potencia, el costo horario deberá estar definido para cada *timeslot* u . De este modo, todas las variables deben estar escaladas para cada *timeslot* y la fórmula del costo de la energía en cada, prc_u , queda definida así:

$$prc_u(s_u) = \begin{cases} a_u, & \text{si } 0 \leq s_u \leq c_u \\ b_u, & \text{si } s_u > c_u \end{cases}$$

Una vez fijado el valor límite de consumo c_u , si el consumo del *timeslot* actual sobrepasa este límite, el precio aplicado será b_u . Este nuevo costo estará definido mediante la siguiente expresión:

$$b_u = \lambda \cdot a_u$$

Donde λ es un valor escalar positivo.

4.3.2. Formulación del problema.

Es necesario que el usuario indique los parámetros de funcionamiento de cada AOA. El primer parámetro es el intervalo de operación (OTI, *operation time interval*) que estará definido por $\alpha_a, \beta_a \in U$ ($\alpha_a < \beta_a$) como los índices de inicio y final. El segundo parámetro es el tiempo de operación (LOT, *length of operation time*) de cada AOA, que estará representado por la variable l_a y se debe cumplir que $\beta_a - \alpha_a \geq l_a$. Mientras más amplio sea el periodo OTI, más opciones podrá encontrar el algoritmo genético.

Otro parámetro dentro del modelamiento es el inicio de operación (OST, *operation start time*) de cada AOA. Este parámetro estará representado por la variable t_a y define el *timeslot* u , dentro del periodo $[\alpha_a, \beta_a]$ donde el AOA iniciará su operación (Zhuang Zhao et al., 2013). Se debe considerar una restricción para esta variable, ya que debe estar dentro del intervalo OTI, y debe permitir una operación completa del AOA. Es decir

$$t_a \in [\alpha_a, \beta_a - l_a]$$

En la Figura 4.8 se muestra un ejemplo del rango de esta variable.

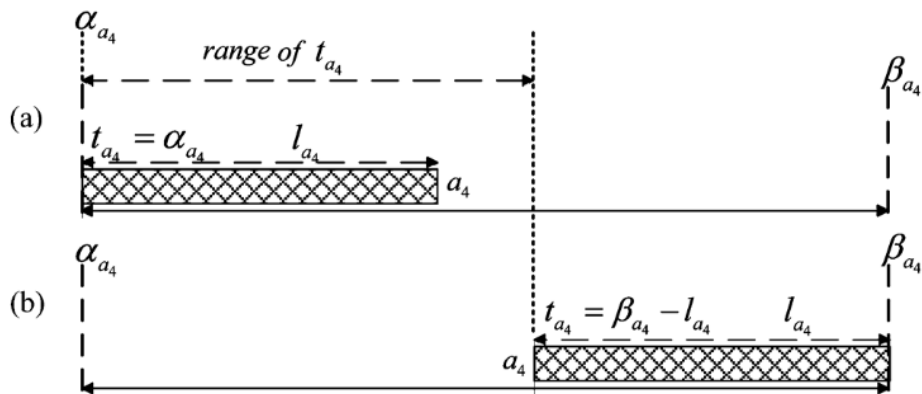


Figura 4.8. Inicio de operación OST: (a) inicio al comienzo del periodo (b) inicio al final del periodo.

En la Figura 4.9 extraída de (Zhuang Zhao et al., 2013), se muestra la relación que existe entre los parámetros que se han definido hasta el momento.

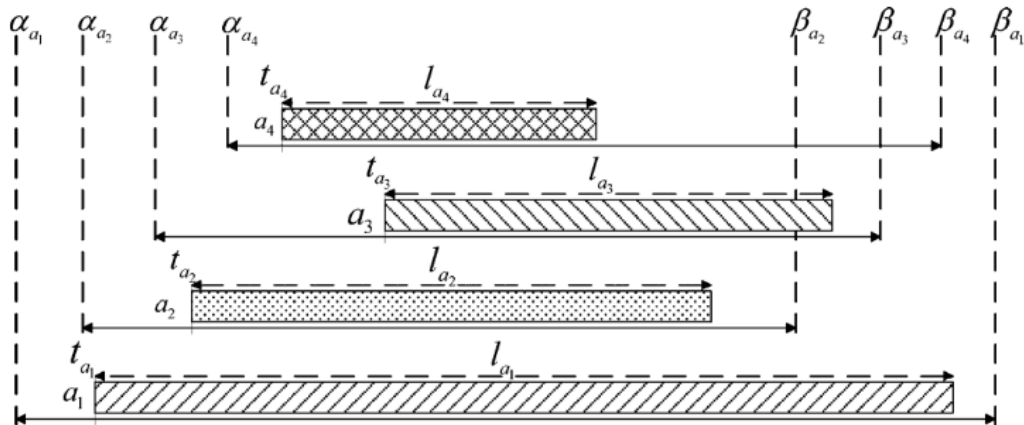


Figura 4.9. Relación de los parámetros de configuración de los AOA.

Hasta este punto, cada AOA está definido por un intervalo de operación OTI $[\alpha_a, \beta_a]$, una duración de operación LOT l_a y la potencia $p_a^{(u)}$. Adicionalmente, se ha definido el inicio de operación OST t_a , variable mediante la cual el algoritmo genético interactuará para definir el cronograma. Utilizando estas variables, se crea la matriz de planificación de consumo de potencia P para todos los AOA.

$$P = \left\{ \begin{array}{ll} p | p_a^{(u)} = \frac{x_a}{5}, & \forall a \in A, u \in [t_a, t_a + l_a] \\ p_a^{(u)} = 0, & \forall a \in A, u \in U \setminus [t_a, t_a + l_a] \end{array} \right\}$$

Considerando que la instalación dispondrá de energía fotovoltaica, se debe incorporar esta variable dentro de la matriz P . Ya que la instalación no dispone de baterías, solo se deberá considerar el caso en que la energía fotovoltaica generada será utilizada mediante los AOA que estén en uso en el instante que esta energía es producida. De esta forma, la instalación fotovoltaica provocará que las horas de generación de energía, donde el costo del servicio sea más elevado (hora punta), sean consideradas como periodos con mayor bondad para el algoritmo genético. Esto significa que, aparte del ahorro de energía consumida al utilizar la energía renovable producida, más AOA podrán cumplir su función lo antes posible.

En el apartado anterior, mediante la herramienta SAM se creó una base de datos con la energía fotovoltaica generada para cada hora a lo largo de un año entero. Sin embargo, se debe tomar en cuenta que la matriz P ha sido modelada para cada *timeslot* u . Al incorporar el término de energía fotovoltaica p_{gen} , la matriz P queda de la siguiente forma:

$$P = \left\{ \begin{array}{ll} p | p_a^{(u)} = \frac{x_a - p_{gen}}{5}, & \forall a \in A, u \in [t_a, t_a + l_a] \\ p_a^{(u)} = 0, & \forall a \in A, u \in U \setminus [t_a, t_a + l_a] \end{array} \right\}$$

En esta matriz, cada fila representa el vector de consumo de un AOA y u el índice de columnas. Sumando todos los elementos de cada columna, es decir, toda la energía consumida por los AOA en cada *timeslot*, se obtiene la siguiente expresión

$$P_{scd} = \{p_{scd} | p_{scd}^{(u)} = \sum P^{(u)}, \forall u \in U\}$$

Donde $P^{(u)}$ representa la columna u de la matriz de planificación de consumo de potencia.

Otro aspecto que se debe considerar es la satisfacción del cliente, que en este caso se traduce en que el usuario quiere que las tareas automáticas se cumplan lo antes posible. Para ello, se introduce una nueva variable que representa el tiempo de retardo de inicio de operación (*DTR, delay time rate*), cuya definición es la siguiente

$$DTR_a = \frac{t_a - \alpha_a}{\beta_a - l_a - \alpha_a}$$

Mediante esta expresión, cuanto más tarde inicie operación el AOA, más grande será el valor DTR que estará dentro del intervalo $[0,1]$ (Zhuang Zhao et al., 2013), es decir, si un determinado artefacto inicia su operación en el primer *timeslot* de su OTI, su DTR valdrá cero, mientras que si inicia su operación en su último instante posible, su DTR valdrá uno.

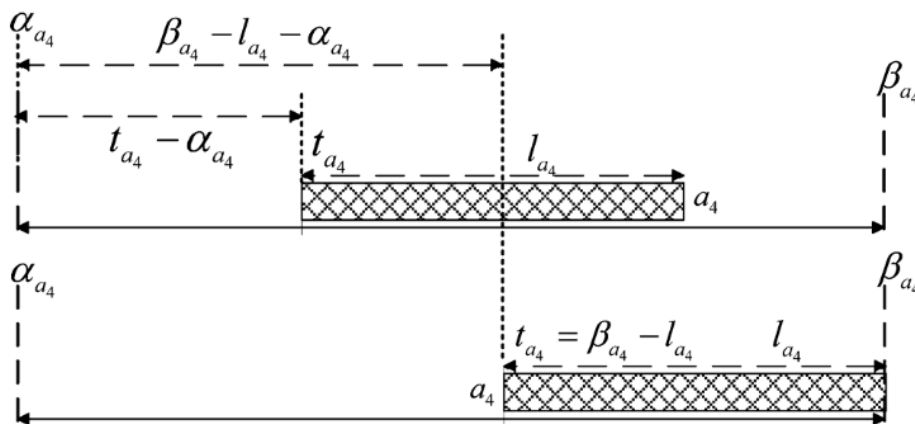


Figura 4.10. Definición del DTR.

Para poder utilizar este valor dentro de la formulación del sistema, se introduce un nuevo parámetro $\rho > 1$, generando una nueva expresión

$$\sum_{a \in A} \rho^{DTR_a}$$

El resultado de la anterior expresión crece conforme el DTR de cada AOA continúa creciendo. De este modo, para los clientes es de interés que este valor sea lo más pequeño posible.

Condensando todo el desarrollo en una expresión, el problema de planificación se resume mediante lo siguiente:

$$\begin{aligned} & \text{minimize } w_1 F_1(P_{scd}) + w_2 F_2(DTR_a) \\ & \text{s. t. } t_a \in [\alpha_a, \beta_a - l_a] \end{aligned}$$

Donde

$$\begin{aligned} F_1(P_{scd}) &= \sum_{u=1}^{120} prc_u(P_{scd}^{(u)}) \cdot P_{scd}^{(u)} \\ F_2(DTR_a) &= \sum_{a \in A} \rho^{DTR_a} \end{aligned}$$

Los términos w_1 y w_2 son los pesos que representan la importancia de cada componente del problema, son valores dentro del intervalo $[0,1]$ y se cumple que $w_1 + w_2 = 1$. El peso w_1 multiplica el costo de la energía mientras que w_2 multiplica el retraso de inicio de operación; esto quiere decir que se debe escoger a que componente se debe dar preferencia en el algoritmo genético haciendo su peso w mayor que el otro. Si se quiere dar un peso igual a los dos factores, se debe hacer que $w_1 = w_2 = 0.5$ (Zhuang Zhao et al., 2013).

Después de la normalización, la formula final del sistema es:

$$\begin{aligned} & \text{minimize } w_1 \cdot \frac{\sum_{u=1}^{120} prc_u(P_{scd}^{(u)}) \cdot P_{scd}^{(u)}}{\max(\sum_{u=1}^{120} prc_u(P_{scd}^{(u)}) \cdot P_{scd}^{(u)})} \\ & \quad + w_2 \cdot \frac{\sum_{a \in A} \rho^{DTR_a}}{\max(\sum_{a \in A} \rho^{DTR_a})} \\ & \text{s. t. } t_a \in [\alpha_a, \beta_a - l_a] \end{aligned}$$

4.4. Aplicación HEMS.

En este apartado se documenta la implementación de la aplicación una vez desarrollados los componentes principales como el modelo matemático, las bases de datos con información necesaria para el algoritmo, entre otros.

Antes de la instalación del proyecto, es importante revisar los conceptos básicos de Laravel y su lenguaje orientado a objetos PHP. Laravel funciona bajo el paradigma Modelo – Vista – Controlador en el cual se establece las vías de comunicación de datos entre los componentes del programa.

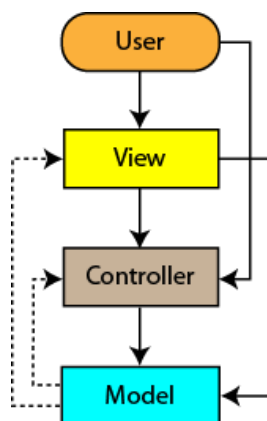


Figura 4.11. Paradigma Modelo-Vista-Controlador.

Controlador

Los controladores son ficheros denominados clases. En estos se programan métodos mediante los cuales se ejecutan acciones de control tales como hacer peticiones a un modelo para solicitar información de la base de datos, enviar información a las vistas o generar eventos.

Vista

Una vista es un fichero encargado de presentar la información recibida de una forma adecuada para que el usuario pueda entenderla. Por ejemplo, los datos pueden ser definidos como una cadena de texto o un arreglo de datos, pero para que sea útil para el usuario deberán ser desplegados como gráficas, tablas, etc.

Modelo

Son ficheros que se encargan de gestionar los accesos a la información de la base de datos. Estos pueden definir privilegios de acceso y reciben las peticiones del controlador.

El desarrollo de la aplicación consiste en programar principalmente los 3 tipos de archivos Modelos, Vistas y Controladores de forma que la interacción de los mismos ejecute la lógica y el comportamiento deseado.

El primer paso consiste en instalar el software previamente seleccionado para empezar con las tareas de programación. Para acelerar el proceso de desarrollo, se ha decidido

utilizar la máquina virtual *Laravel Homestead* que es una herramienta oficial para el desarrollo de aplicaciones con *Laravel*. Esta máquina virtual viene pre cargada con el sistema operativo *Linux Ubuntu* y todos los programas necesarios para completar el entorno de desarrollo local como Composer (gestor de paquetes), PHP (lenguaje de programación), Nginx (servidor web), MySQL (base de datos), entre otros (Laravel, s. f.-b).

Después que se ha instalado y configurado *Laravel Homestead*, se procede a crear el proyecto mediante la interfaz de comandos. Para esto se introduce el siguiente texto:

```
composer create-project --prefer-dist laravel/laravel homenergymanager
```

Una vez creado el proyecto que se ha denominado *Homenergymanager*, se abre esta carpeta en el editor de texto de preferencia. En este caso se usa *VSCoDe* por ser el software más conocido por el autor. En la Figura 4.12 se puede observar el directorio raíz del proyecto creado.

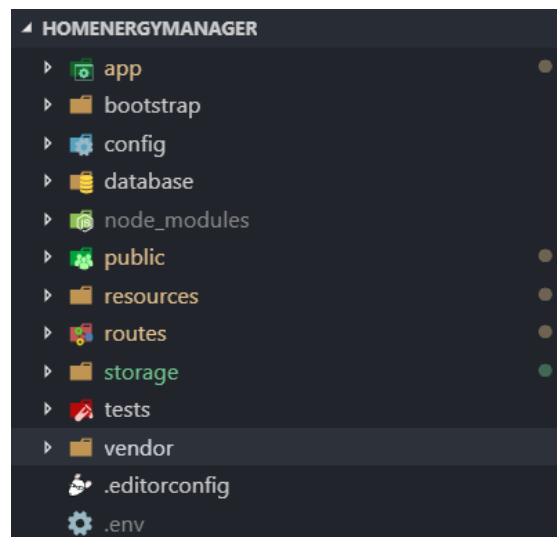


Figura 4.12. Directorio raíz del proyecto Homenergymanager.

Como puntos importantes, se deben destacar ciertos subdirectorios con los cuales se trabajará constantemente durante el desarrollo como la ruta */app/Http/Controllers* que es donde se alojarán los ficheros que sirven como controladores para el manejo de los diferentes recursos.

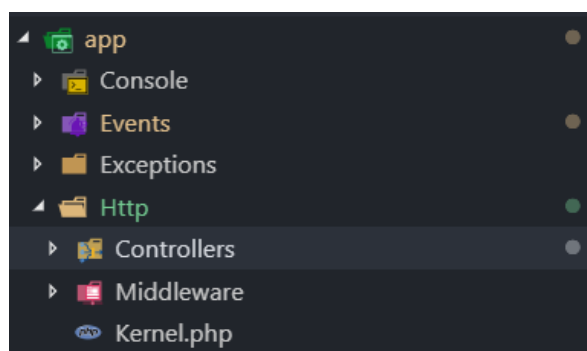


Figura 4.13. Directorio de la carpeta Controllers.

Otra ruta de común acceso es **/resources**. Dentro de esta dirección se encuentran las carpetas *js* (componentes interactivos en lenguaje javascript), *sass* (código que da estilos y colores a los componentes de la aplicación) y *views* (código html de las páginas web o ficheros parciales).

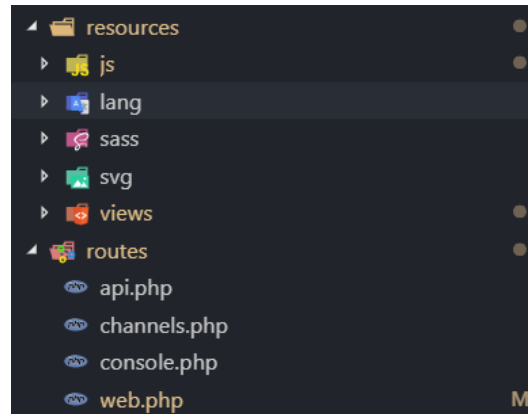


Figura 4.14. Directorio de las carpetas js, sass y views.

4.4.1. Configuraciones básicas de Laravel

Después de que se ha instalado tanto el software como el proyecto correctamente, el siguiente paso consiste en realizar unas configuraciones iniciales de forma que toda la lógica programada pueda ser ejecutada y mostrada en el navegador web o en la interfaz de comandos.

Base de datos

Es necesario crear una base de datos donde se crearán y almacenarán tablas con la información de interés para el desarrollo de la aplicación HEMS. Se ha utilizado el programa *phpMyAdmin* que funciona en un navegador web, para administrar las bases de datos. En la Figura 4.15 se muestra la ventana donde se crea la base de datos denominada “**hems**”.

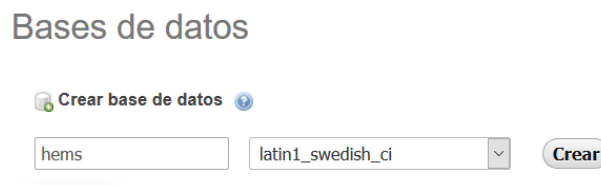


Figura 4.15. Ventana de creación de base de datos en phpMyAdmin.

Luego, para poder escribir/leer información de esta base de datos, es necesario indicarle al proyecto las credenciales dentro del archivo **.env** que se encuentra en el directorio raíz de la aplicación (ver Figura 4.16). En este archivo se alojan las variables de configuración de la aplicación como por ejemplo credenciales de base de datos, configuración del servicio de mail, credenciales del servicio de *broadcasting*, entre otros.

```
9 DB_CONNECTION=mysql
10 DB_HOST=127.0.0.1
11 DB_PORT=3306
12 DB_DATABASE=hems
13 DB_USERNAME=homestead
14 DB_PASSWORD=secret
```

Figura 4.16. Configuración de la base de datos en el archivo .env

Migraciones

Son archivos donde se define la estructura de la tabla (base de datos) de información que será almacenada. Por ejemplo, si una aplicación permite el registro de usuarios, existirá una migración que defina los campos que debe tener cada usuario dentro de la base de datos. Inicialmente solo existirá la migración para los usuarios, sin embargo, durante el desarrollo se irán generando más migraciones para los recursos necesarios como los AOA, la producción y costo de energía diaria, entre otros.

Al iniciar una nueva aplicación, es importante migrar los esquemas (o migraciones) hacia la base de datos para poder empezar a probar la aplicación escribiendo directamente en ella, y cada vez que se cree un nuevo modelo se deberá realizar la migración del mismo.

La acción de migrar un esquema quiere decir que se crea la tabla en la base de datos considerando los campos de información definidos en la migración. Esta tarea es realizada de forma automática por el entorno Laravel mediante el comando `php artisan migrate`. Utilizar las migraciones supone una ventaja para los desarrolladores ya que se trabajan los esquemas de información a nivel de código, además que facilita la modificación de la estructura de la tabla de información.

Autenticación

Debido a que en el desarrollo de este proyecto se permite el registro de nuevos usuarios, es necesario crear la lógica para gestionar el inicio de sesión, así como las nuevas altas en el sistema. Sin embargo, Laravel ofrece esta configuración de forma instantánea al utilizar el comando `php artisan make:auth`.

4.4.2. Algoritmo Genético

El desarrollo del algoritmo genético es una tarea extensa y requiere de conocimientos intermedio/avanzado en programación orientada a objetos. Una vez que se ha entendido y modelado el problema a tratar en el presente proyecto, se ha creado un subdirectorío específicamente para el funcionamiento de este algoritmo dentro de los servicios registrados en el proyecto *Homenergymanager*. La ruta de estos archivos es `/app/Services/GeneticAlgorithm` que se muestra en la Figura 4.17

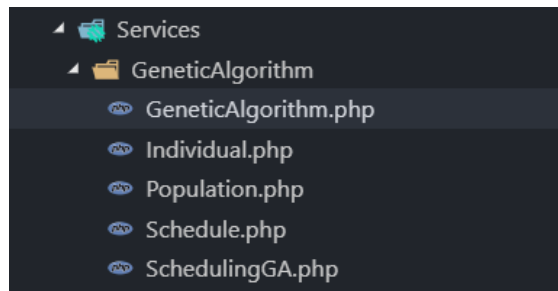


Figura 4.17. Directorio del algoritmo genético.

En este directorio se pueden observar varios archivos. Ya que el lenguaje PHP es programación orientada a objetos, se han creado varios ficheros o clases para manejar cada parte del algoritmo genético, además de la clase donde se controla todo el proceso.

La clase **Individual.php** representa un individuo dentro del algoritmo genético, es decir, este porta la información de una potencial solución (cronograma de funcionamiento de los AOAs) dentro de la población. Además, es una clase independiente (no depende de otros ficheros) donde se han declarado las propiedades y métodos descritos en la Tabla 4.4

Tabla 4.4. Propiedades y métodos de la Clase “Individual”.

	Nombre	Tipo	Descripción
Propiedades	\$chromosome	array	Porta el contenido de la potencial solución, es decir, el cronograma de funcionamiento de los AOAs.
	\$fitness	double	Contiene el valor fitness de la solución.
Métodos	random	public static function	Retorna una nueva solución aleatoria.
	getChromosome	public function	Devuelve la propiedad \$chromosome.
	setGene	public function	Modifica un gen de la propiedad \$chromosome.
	getGene	public function	Retorna un gen de la propiedad \$chromosome.
	setFitness	public function	Modifica la propiedad \$fitness de la solución.
	getFitness	public function	Devuelve la propiedad \$fitness de la solución.

Después se tiene la clase **Population.php**. Esta clase representa toda la población de soluciones. De hecho, cada solución de la población es una instancia de la clase *Individual* revisada anteriormente, lo cual indica que cada solución tendrá los mismos métodos y propiedades con los que el algoritmo interactúa. Por su parte, la clase “Population” está compuesta por los métodos y propiedades indicados en la Tabla 4.5 que corresponden a datos relacionados con toda la población de individuos.

Tabla 4.5. Propiedades y métodos de la Clase "Population".

	Nombre	Tipo	Descripción
Propiedades	\$population	array	Contiene un conjunto de soluciones (cronograma de funcionamiento de AOA)
	\$populationFitness	double	Contiene el valor fitness global de la población.
Métodos	random	public static function	Retorna una nueva población aleatoria.
	getIndividuals	public function	Devuelve la propiedad \$population.
	getFittest	public function	Devuelve la solución con mejor fitness.
	setPopulationFitness	public function	Modifica la propiedad \$populationFitness
	getPopulationFitness	public function	Retorna el valor de la propiedad \$populationFitness.
	setIndividual	public function	Modifica una solución dentro de la población.
	getIndividual	public function	Devuelve una solución dentro de la población.
	shuffle	public function	Ordena la población de forma aleatoria.
	getAvgFitness	public function	Retorna el valor fitness promedio de la población.

La siguiente clase *Schedule.php*. Esta clase porta todas las propiedades y métodos que describen el cronograma que se busca como solución.

Tabla 4.6. Métodos y propiedades de la Clase "Schedule".

	Nombre	Tipo	Descripción
Propiedades	\$energyCost	array	Almacena los valores del costo horario de la energía.
	\$energyPV	array	Almacena los valores de la producción horaria de energía.
	\$appliances	array	Contiene la información de los artefactos del hogar y sus características de funcionamiento.
	\$appliancesCount	integer	Contiene el número de artefactos dentro del sistema.
	\$timeSlots	integer	Contiene el número de timeslots del día
Métodos	setAppliances	public function	Modifica la propiedad \$appliances.
	setEnergyCost	public function	Modifica la propiedad \$energyCost.
	setEnergyPV	public function	Modifica la propiedad \$energyPV.
	setAppliancesCount	public function	Modifica la propiedad \$appliancesCount.
	getAppliances	public function	Devuelve el array \$appliances.
	getEnergyCost	public function	Devuelve el array \$energyCost.
	getEnergyPV	public function	Devuelve el array \$energyPV.
	getAppliancesCount	public function	Retorna el número de artefactos.
	getTimeSlots	public function	Retorna el número de timeslots

Cuando se crea una instancia de *Schedule*, este objeto adquiere todas estas características para irse modificando a lo largo del algoritmo genético y que al final presentara la solución más óptima encontrada. El objeto *Schedule* contará con las propiedades y métodos que se detallan en la Tabla 4.6.

Cabe destacar que la clase *Schedule* trabaja con las variables desarrolladas en los numerales 4.1 y 4.2 referentes a la generación de energía fotovoltaica y al análisis dinámico del costo de la energía eléctrica. Además, al igual que las clases “*Individual*” y “*Population*”, *Schedule* también es un archivo independiente que no requiere de otros métodos externos.

La clase ***GeneticAlgorithm.php*** es la encargada de realizar todas las operaciones concernientes al algoritmo previamente estudiado. Este objeto se alimenta de la información que se le proporciona sobre los artefactos, individuos y población para realizar las operaciones necesarias como selección, cruce o mutación y así desarrollar la solución más óptima. En esta clase tiene las propiedades y métodos descritos en la Tabla 4.7.

Tabla 4.7. Propiedades y métodos de la Clase “*GeneticAlgorithm*”.

	Nombre	Tipo	Descripción
Propiedades	\$populationSize	integer	Almacena el número de soluciones dentro de la población.
	\$mutationRate	double	Almacena la proporción de mutación.
	\$crossoverRate	double	Almacena la proporción de cruce
	\$elitismCount	integer	Almacena la cuenta de individuos elites
	\$tournamentSize	integer	Almacena el tamaño del torneo para la selección.
	\$temperature	integer	Representa la probabilidad de mutación.
	\$coolingRate	integer	Almacena la proporción de eliminación de mutación.
Métodos	initPopulation	public function	Establece una población inicial aleatoria.
	getTemperature	public function	Retorna el valor de \$temperature
	coolTemperature	public function	Disminuye el valor de \$temperature restando el \$coolingRate
	calculateFitness	public function	Calcula el valor fitness de una solución o Individual
	evaluatePopulation	public function	Evalúa todas las soluciones de la población mediante un bucle
	terminationCondition	public function	Verifica si se cumple la condición de finalización.
	generationMaxedOut	public function	Verifica si se han sobrepasado el máximo de generaciones
	selectParent	public function	Selecciona una solución Padre dentro de la población para las operaciones
	crossoverPopulation	public function	Realiza el cruce de la población
	mutatePopulation	public function	Realiza la mutación de la población

Finalmente, se la clase *SchedulingGA.php*. Este es el encargado de controlar todo el proceso de funcionamiento del algoritmo genético desde que se inician las iteraciones, hasta que se selecciona la solución más óptima. Esta clase, a diferencia de las anteriores, depende de las otras clases y además necesita saber la forma de como solicitar información de la base de datos. Es por eso que en la cabecera de esta clase se debe indicar los archivos que necesita para ejecutar las acciones programadas mediante código. La Figura 4.18 muestra las rutas de los archivos que serán importados.

```

1  <?php
2
3  namespace App\Services\GeneticAlgorithm;
4
5  use App\Events\ScheduleGenerated;
6  use App\Schedule as ScheduleModel;
7  use App\Appliance as ApplianceModel;
8  use Carbon\Carbon;
9  use App\Prcu;
10 use App\PowerGenerated;
11
12 class SchedulingGA
13 { ...
182 }

```

Figura 4.18. Cabecera de la clase SchedulingGA.

En la Tabla 4.8 se muestran las propiedades y métodos de la Clase “SchedulingGA”

Tabla 4.8. Propiedades y métodos de la Clase “SchedulingGA”

	Nombre	Tipo	Descripción
Propiedad	\$schedule	array	Almacena los valores y características del cronograma generado
Métodos	getPV	public function	Retorna un array con la energía horaria generada almacenada en la base de datos
	getEnergyCost	public function	Retorna un array con el costo de la energía horaria alojada en la base de datos
	initializeSchedule	public function	Inicializa el objeto Schedule con la información de costo de energía, energía generada y artefactos.
	run	public function	Realiza el bucle del algoritmo genético.

Se ha decidido no incluir en este reporte, líneas de código de programación ya que, por su gran extensión y combinación de archivos, llevaría a un grado bajo de entendimiento del desarrollo del presente proyecto. Para un mejor entendimiento del funcionamiento del algoritmo genético implementado, se ha realizado el diagrama de flujo que describe toda la programación desarrollada en las clases anteriormente descritos.

En (Althaher et al., 2015), el funcionamiento del algoritmo genético es denominado motor de optimización. En la Figura 4.19 obtenida de dicha investigación, se puede observar el diagrama de flujo ideal para una aplicación automática HEMS donde se interactúa con variables externas, que en este caso en particular será representado con tablas previamente generadas y almacenadas en una base de datos.

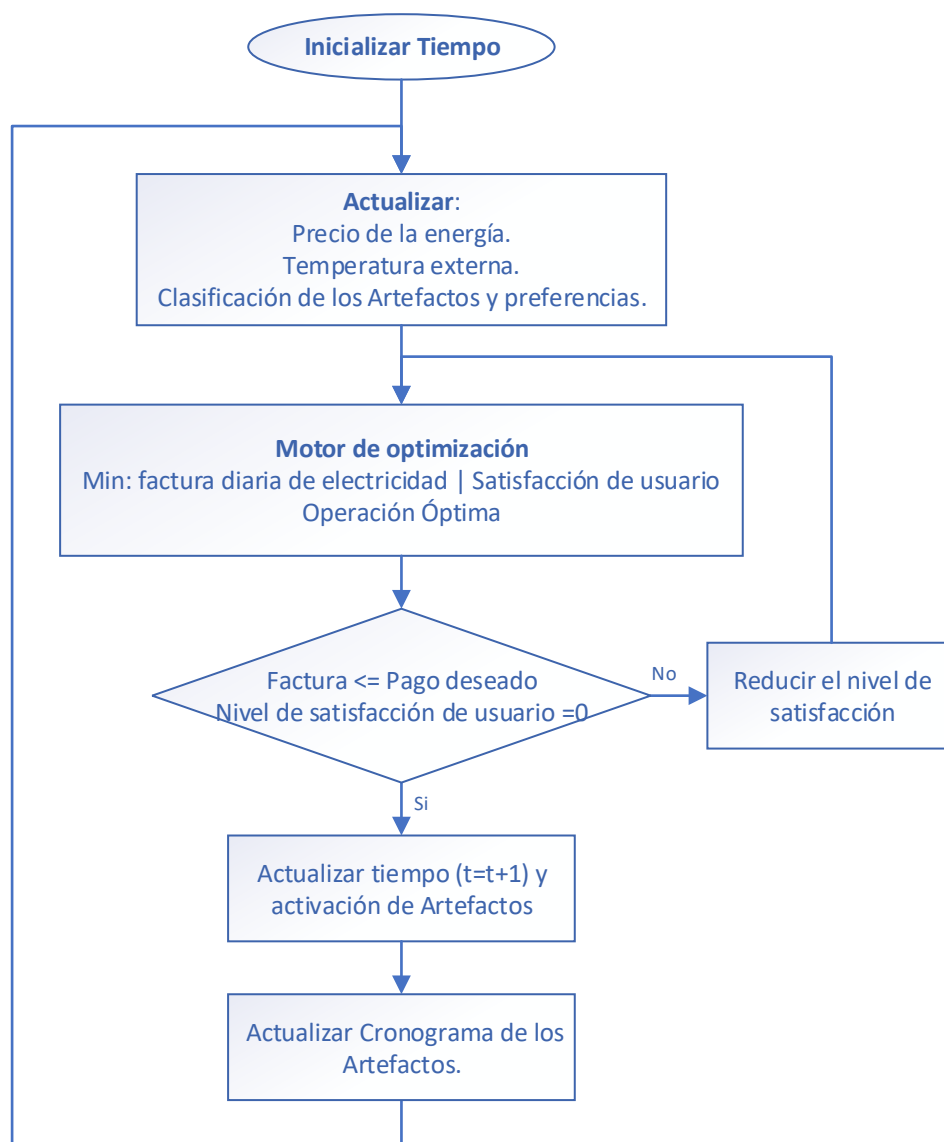


Figura 4.19. Diagrama de flujo ideal de un sistema ADR (automated demand response).

El motor de optimización antes mencionado es donde reside el algoritmo genético, que es el algoritmo de optimización desarrollado como parte fundamental de la aplicación HEMS del presente proyecto.

Las clases *Individual*, *Population*, *Schedule* y *GeneticAlgorithm* previamente explicados, interactúan entre si mediante el flujo que marca la clase *SchedulingGA*. De esta forma se consolida un solo servicio llamado *GeneticAlgorithm* que, de forma iterativa, genera un sinnúmero de soluciones posibles para finalmente entregar un cronograma de funcionamiento de los artefactos que optimice el consumo de la energía, y, por ende, minimice el costo de esta. Todo este proceso se detalla en los diagramas de flujo de la Figura 4.20 y la Figura 4.21.

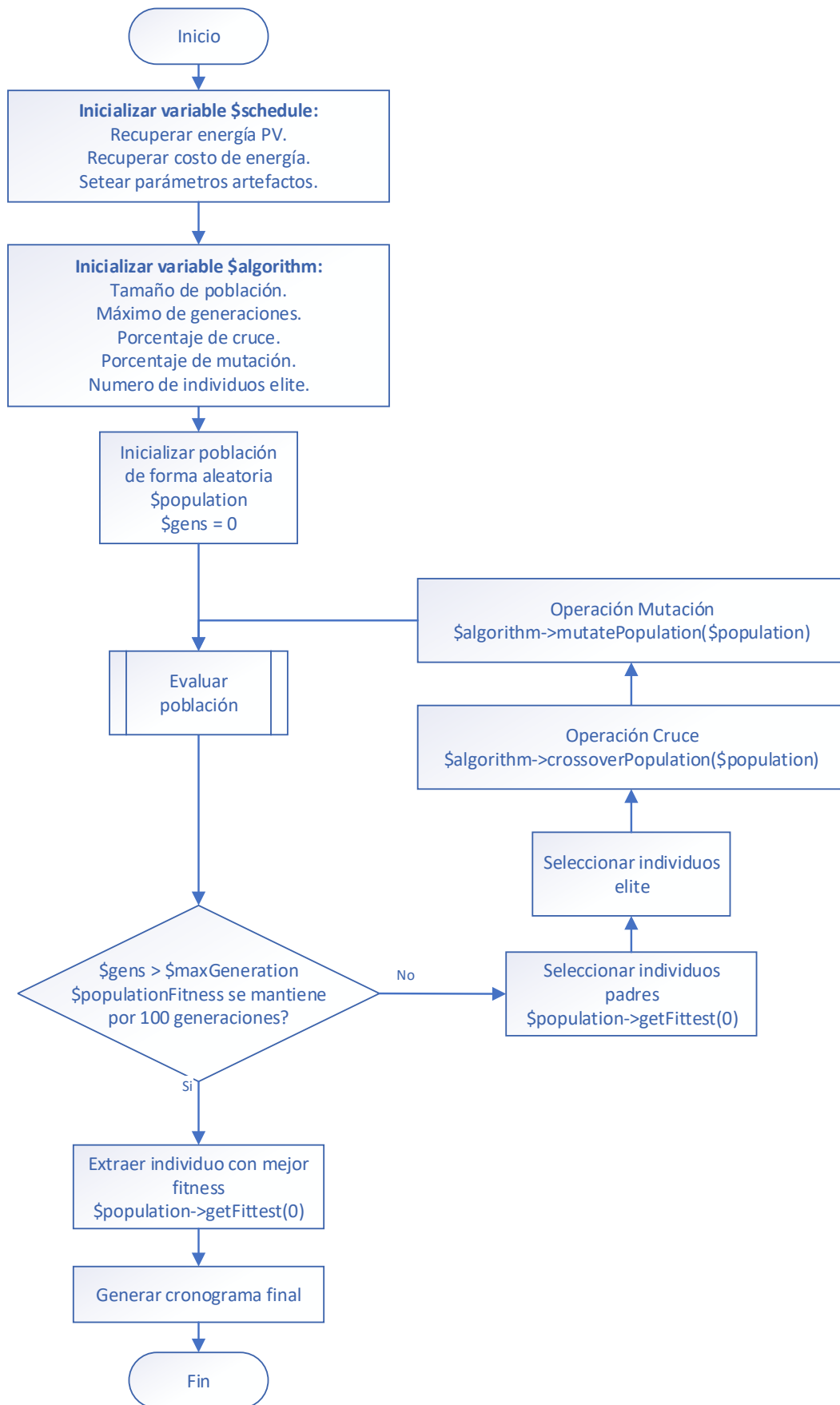


Figura 4.20. Diagrama de flujo del algoritmo genético.

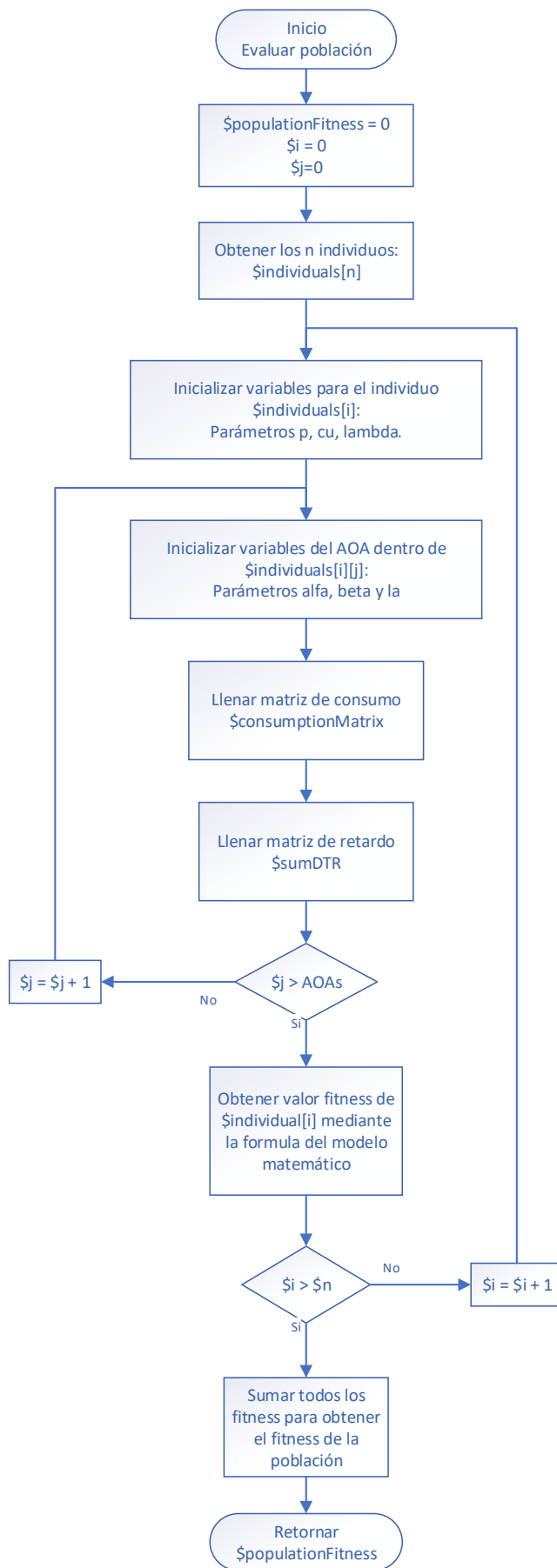


Figura 4.21. Diagrama de flujo del subproceso "Evaluar Población".

4.4.3. Parámetros de funcionamiento.

Para poder simular el algoritmo genético, hay que definir algunas variables que han surgido a lo largo del desarrollo del mismo. Estas variables están presentes dentro del modelo matemático y, por ende, del algoritmo genético, y su desempeño depende directamente de estos. Los valores de estos parámetros son los que se muestran en la Tabla 4.9, obtenida de (Zhuang Zhao et al., 2013).

Tabla 4.9. Parámetros de funcionamiento del HEMS y Algoritmo Genético.

Modelo Matemático		
Descripción	Símbolo	Valor
Número de artefactos instalados en el hogar	n_a	16
Límite de consumo eléctrico por cada <i>timeslot</i> *	c_u	0,4 kWh
Factor de ajuste del costo de energía al sobrepasar el límite de consumo en un <i>timeslot</i> *	λ	1,4423
Parámetro de retardo*	ρ	5
Algoritmo Genético		
Descripción	Símbolo	Valor
Tamaño de población	N_s	200
Probabilidad de cruce	P_c	0,9
Probabilidad de mutación	P_m	0,02
Generaciones máximas	N_{gmax}	1000

* variables del modelo matemático. Capítulo 4.3

De la misma forma, la Tabla 4.10 también extraída de (Zhuang Zhao et al., 2013), muestra un conjunto de AOA con sus preferencias de usuario que servirán para poder correr el algoritmo y realizar las pruebas necesarias.

Tabla 4.10. Artefactos automáticos AOA y sus preferencias de usuario.

AOA*	OTI* - Inicio	OTI* - Fin	LOT*	Potencia [kW]
Aire acondicionado 1	41	60	5	1
Aire acondicionado 2	61	85	5	1
Aire acondicionado 3	86	120	10	1
Radiador eléctrico 1	1	30	5	1,8
Radiador eléctrico 2	91	115	10	1,8
Olla arrocera 1	1	25	2	0,5
Olla arrocera 2	41	60	2	0,5
Olla arrocera 3	71	90	2	0,5
Calentador de agua	81	105	3	1,5
Lavaplatos	101	120	2	0,6
Lavadora de ropa	1	60	5	0,38
Tetera eléctrica 1	1	25	1	1,5
Tetera eléctrica 2	66	85	1	1,5
Humidificador 1	1	30	10	0,05
Humidificador 2	91	120	10	0,05
Secadora de ropa	30	40	5	0,8

* ver el glosario de términos

5. INTERFAZ DE USUARIO

En este apartado se muestra el desarrollo de la interfaz de usuario y todos los elementos que lo componen. Además, se describen las funcionalidades básicas que cualquier usuario de la plataforma puede acceder.

Al ingresar a la aplicación, la primera vista que se obtiene es la que se muestra la Figura 5.1. En esta se pueden destacar dos partes principales. La primera, resaltada en el recuadro rojo, son las opciones de *Iniciar sesión* y *Crear cuenta*. Estas opciones permiten el ingreso al sistema de un usuario registrado previamente o el registro de un nuevo usuario respectivamente. En el recuadro azul por otra parte se presentan 4 opciones que dirigen al usuario a las ventanas de funcionamiento de la aplicación como *Dashboard*, *Histórico*, *Gráficas* y finalmente *Mi instalación*.

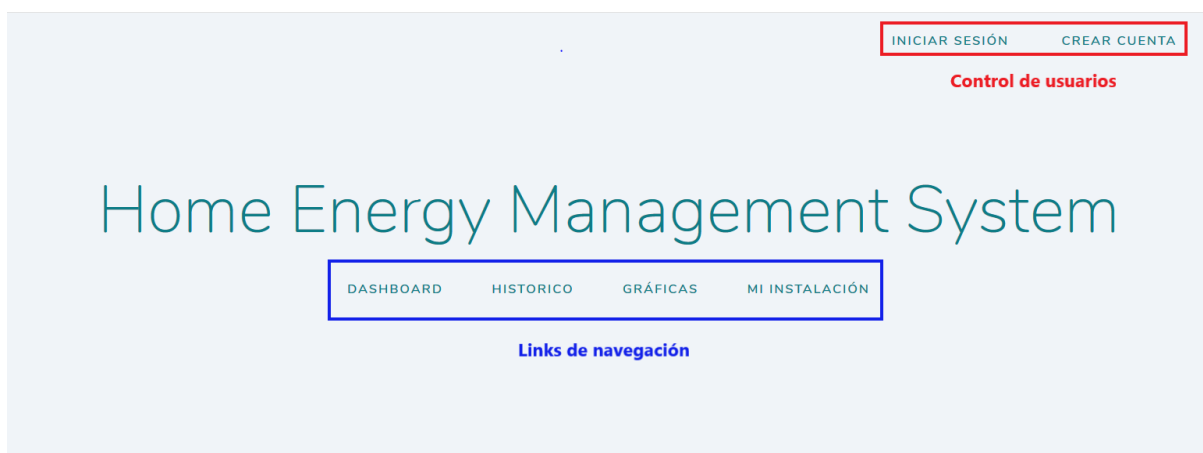


Figura 5.1. Ventana de bienvenida a la aplicación HEMS.

5.1. Funciones básicas.

5.1.1. Ingreso de usuario

La aplicación HEMS desarrollada permite el registro de usuarios donde cada uno de ellos puede administrar su instalación, verificar gráficas y datos históricos, entre otros. Cuando el usuario es nuevo, se debe registrar mediante la opción *Crear cuenta*. Esa acción lo llevará a la ventana indicada en la Figura 5.2 donde se deben completar los campos *Nombre*, *Email* y *Contraseña*.

Por otra parte, si el usuario ya se ha registrado anteriormente, este debe ingresar mediante la opción de *Iniciar sesión*. Esta opción requiere el email y la contraseña con el que se registró el usuario, tal y como indica la Figura 5.3.



Figura 5.2. Ventana de registro de nuevo usuario.

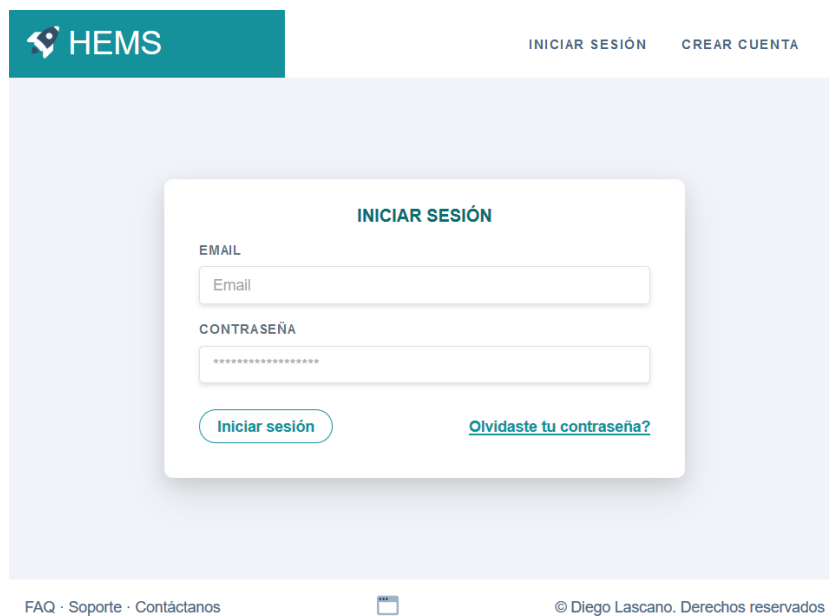


Figura 5.3. Ventana de ingreso de usuario previamente registrado.

Después de que el usuario ha ingresado al sistema ya sea mediante *Iniciar sesión* o *Crear cuenta*, la aplicación lo redirecciona a la ventana principal donde ya se pueden observar más opciones de funcionamiento.

5.1.2. Pantallas de funcionamiento.

La primera pantalla de ingreso es la que se muestra en la Figura 5.4. En el recuadro de color naranja se muestran las opciones de navegación, en color verde se muestran las opciones de visualización de los datos, y en recuadro rojo se pueden visualizar los mensajes de advertencia o informativos sobre las acciones en curso de la aplicación.

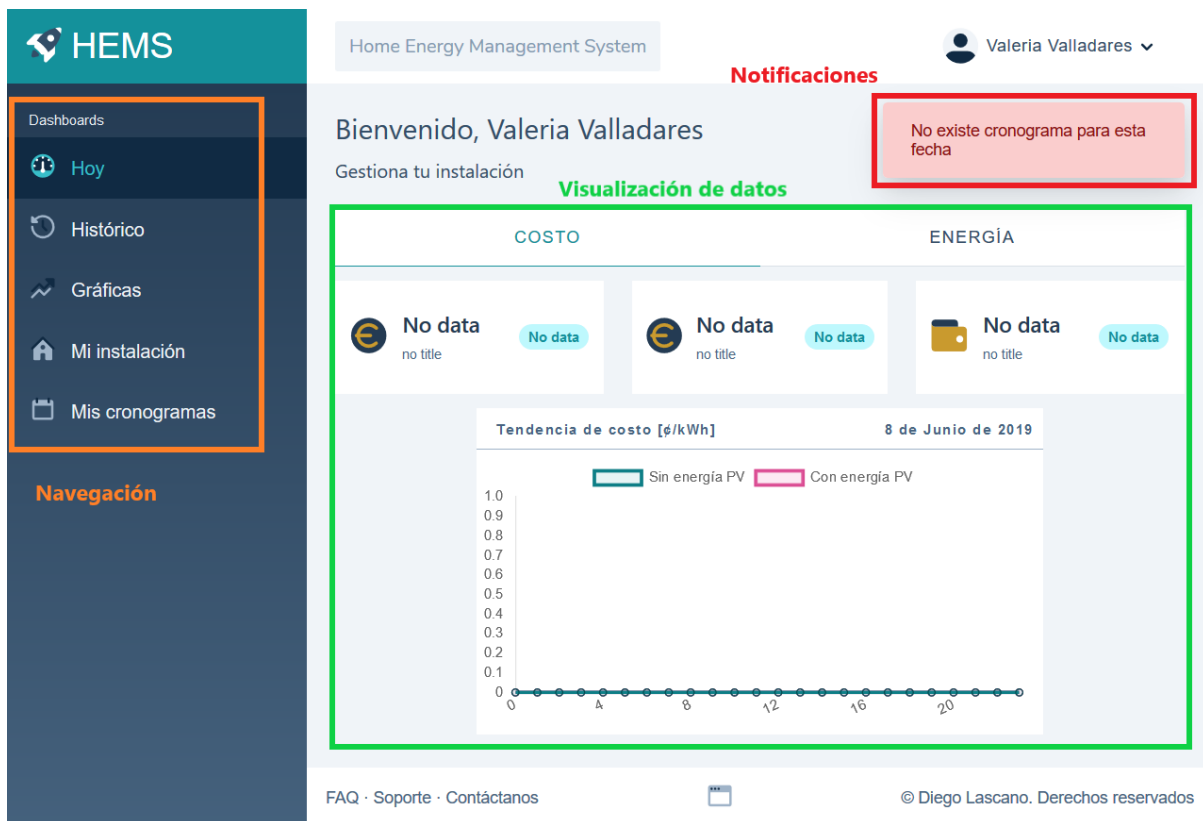


Figura 5.4. Pantalla de inicio de la aplicación.

En las opciones de navegación se muestran 5 opciones. La primera opción, *Hoy*, y la segunda, *Histórico*, muestran el mismo diseño e información. La diferencia radica en que en la pestaña *Hoy*, los datos son exclusivamente al día actual, y los componentes de la pantalla responden a cambios en tiempo real, mientras que, en la pestaña *Historial*, los datos se actualizan según la fecha seleccionada en el calendario.

La Figura 5.5 muestra los componentes principales de las ventanas mencionadas. En el recuadro violeta se muestra un gráfico del día completo donde se muestra el consumo de energía obtenido mediante la simulación del software SAM y el consumo real. En las tarjetas dentro del recuadro verde, se muestran valores promedio de todo el día. Tanto los valores de las tarjetas como los del gráfico pueden ser modificados mediante las opciones seleccionables dentro del recuadro rojo; de esta forma se puede visualizar la información en términos de costo (valores expresados en €) o en términos de energía (valores expresados en kWh).



Figura 5.5. Despliegue de datos de las ventanas Histórico y Hoy.

Las tarjetas de resumen diario (Figura 5.6) muestran un valor promedio de todo el día. Por ejemplo, en términos de costo, estas pueden presentar el valor gastado de todo el día, considerando el uso de energía fotovoltaica, o el consumo bruto por parte de los artefactos. Estas tarjetas muestran 3 secciones: la primera es un ícono que permite al usuario identificar rápidamente de qué se trata la tarjeta; por ejemplo, el símbolo del euro (€) para los casos que se trata de valores económicos. La segunda sección ubicada en el centro de la tarjeta es el valor numérico acompañado del nombre de la variable, y la tercera ubicada a la derecha muestra un valor de porcentaje respecto al valor obtenido al día anterior; este valor se muestra de color verde cuando el valor obtenido es favorable o en color rojo cuando la variable ha cambiado de forma no favorable.

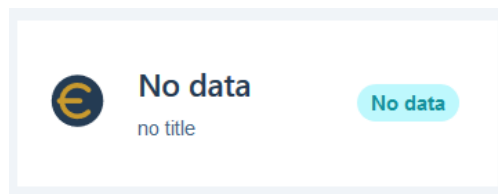


Figura 5.6. Tarjeta de resumen diario.

La siguiente ventana es la denominada Gráficas. En esta se pueden realizar comparaciones de una variable considerando dos días distintos. La Figura 5.7 muestra la disposición de los componentes de esta ventana. En el recuadro azul, el usuario puede seleccionar que variable quiere analizar; de esta forma se tienen tres opciones que son: energía fotovoltaica simulada (datos obtenidos mediante la aplicación SAM), energía fotovoltaica real y finalmente el costo de la energía eléctrica. Cabe indicar que para este proyecto no se ha incorporado ningún hardware que permita obtener datos de energía fotovoltaica real producida, por lo que esta base de datos ha sido generada a partir de la base de datos de energía fotovoltaica simulada, introduciendo un desfase en los días para que los datos sean diferentes, pero dentro de lo esperado.

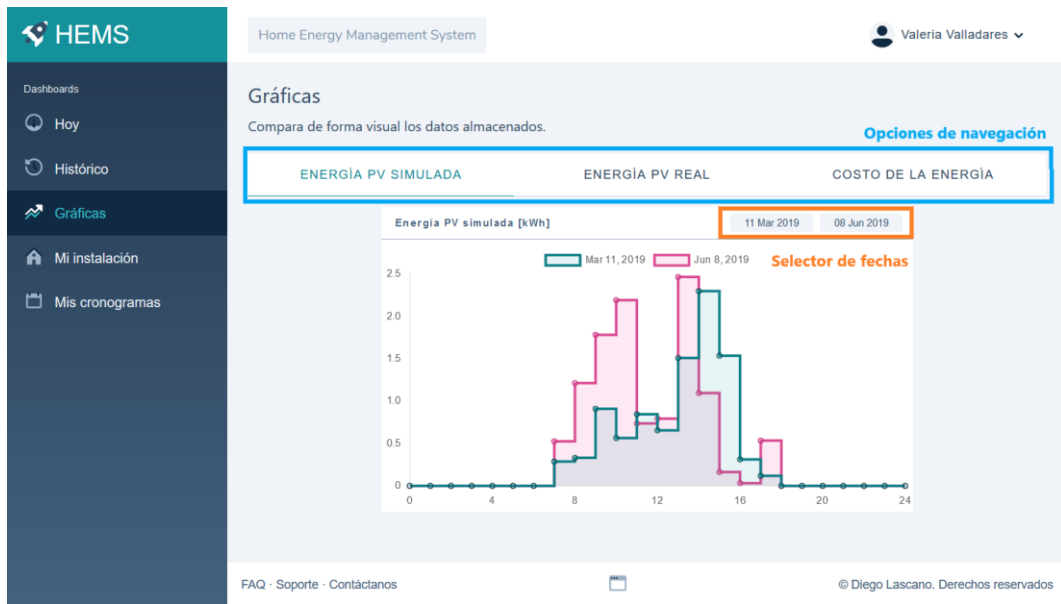


Figura 5.7. Ventana Gráficas, para comparaciones de variables en la base de datos.

Una vez que se ha seleccionado la variable que se quiere analizar dentro de las 3 opciones, en los campos de fecha dentro del recuadro naranja se deben indicar los días de interés. Al dar *click* en cada fecha se desplegará un calendario para poder seleccionar el día de forma más fácil.

5.2. Artefactos automáticos del hogar (AOA).

La ventana *Mi instalación* muestra los artefactos automáticos instalados en el hogar. Estos deben ser registrados por cada usuario cuando se crea un nuevo perfil. La Figura 5.8 muestra los componentes de esta pestaña de navegación. En ella se puede observar una tabla donde se despliega la información referente a cada AOA dentro de la instalación. Esta información indica el nombre, potencia y parámetros de funcionamiento del artefacto como inicio y fin del rango de operación, la duración del ciclo de funcionamiento y el estado del mismo.

ARTEFACTOS	« Previous	Next »	Crear nuevo artefacto		
NOMBRE	POTENCIA[KWH]	INICIO[H]	FIN[H]	DURACIÓN[H]	ESTADO
Lavadora	0.380	1	12	60	0
Tetera eléctrica	1.500	1	5	12	0
Tetera eléctrica	1.500	14	17	12	0
Humidificador	0.050	1	6	120	0
Humidificador	0.050	19	24	120	0
Secadora	0.800	6	8	60	0

Figura 5.8. Ventana Mi instalación. AOA dentro de la instalación del usuario.

Mediante el botón *Crear nuevo artefacto*, el usuario accede a un formulario donde debe ingresar la información técnica de cada artefacto, tal y como se indica en la Figura 5.9. Las horas de inicio y fin indican el rango dentro del cual el algoritmo genético seleccionará la hora efectiva de inicio de operación del artefacto automático.

The screenshot shows the 'NUEVO ARTEFACTO' form in the HEMS interface. The form is titled 'NUEVO ARTEFACTO' and contains the following fields and values:

- ARTEFACTO:** Aire acondicionado
- POTENCIA [KW]:** 1.2
- INICIO:** 1:00
- FIN:** 1:00
- DURACION [MIN]:** 30

Buttons for 'Aceptar' and 'Cancelar' are located at the bottom of the form. The interface also shows a sidebar with navigation options like 'Dashboards', 'Hoy', 'Histórico', 'Gráficas', 'Mi instalación', and 'Mis cronogramas'.

Figura 5.9. Ventana de ingreso de un nuevo AOA.

Después de ingresar todos los artefactos, la tabla de AOA se muestra como en la Figura 5.10. En ella se puede ver que cada artefacto creado ha sido almacenado con todos los valores ingresados en el formulario de creación. Cabe indicar que las variables de tipo tiempo se almacenan en la base de datos en función de *timeslots* para que el cálculo del cronograma del algoritmo sea más fácil.

The screenshot shows the 'Mi Instalación' page in the HEMS interface. A notification 'Nuevo artefacto creado exitosamente' is displayed at the top right. The main content is a table titled 'ARTEFACTOS' with the following data:

NOMBRE	POTENCIA[KWH]	INICIO[H]	FIN[H]	DURACIÓN[H]	ESTADO
Aire acondicionado	1.000	9	12	60	0
Aire acondicionado	1.000	13	17	60	0
Aire acondicionado	1.000	18	24	120	0
Radiador eléctrico	1.800	1	6	60	0
Radiador eléctrico	1.800	19	23	120	0
Olla arrocera	0.500	1	5	24	0
Olla arrocera	0.500	9	12	24	0
Olla arrocera	0.500	15	18	24	0
Calentador de agua	1.500	17	21	36	0
Lavaplatos	0.600	21	24	24	0

The interface also shows a sidebar with navigation options and a footer with 'FAQ · Soporte · Contáctanos' and '© Diego Lascano. Derechos reservados'.

Figura 5.10. Ventana Mi instalación con los artefactos automáticos registrados.

Si el usuario requiere realizar alguna modificación a uno de los AOA's o incluso eliminarlo, se debe dar un *click* sobre el nombre del artefacto. Esta acción lo llevará a la ventana que se muestra en la Figura 5.11 donde se puede seleccionar entre las opciones *Editar* y *Borrar*.



Figura 5.11. Ventana de acciones de control para los AOA's.

5.3. Cronogramas de optimización.

En lo que respecta a la generación de cronogramas de optimización, se ha incluido una opción de navegación denominada *Mis cronogramas*. En ella se puede observar una tabla similar a la de *Mi instalación* donde en esta ocasión se incluye información proporcionada por el algoritmo genético. Cada cronograma generado muestra un ID, el cromosoma con la información del cronograma, el valor fitness obtenido durante el proceso del algoritmo, el número de generaciones evaluadas, además de la fecha de generación (ver Figura 5.12)



Figura 5.12. Ventana Mis cronogramas con los cronogramas generados por el usuario actual.

6. PRUEBAS Y RESULTADOS

En el siguiente apartado se muestran tanto las pruebas de funcionamiento aplicadas al sistema HEMS desarrollado, como los resultados y desempeño obtenido durante estas pruebas.

6.1. Pruebas.

Para la etapa de pruebas y funcionamiento, el algoritmo genético requiere que se hayan cumplido unos requisitos previos de carácter informativo y de configuración. Por una parte, se debe considerar que la información de los AOAs esté completa, esto quiere decir que todos los aspectos técnicos referentes a los artefactos automáticos deben haber sido registrados y validados por el usuario; aspectos como el rango de funcionamiento, duración del ciclo, entre otras. Para cumplir esta parte, dentro del usuario registrado como Valeria Valladares se han ingresado los valores de la Tabla 4.10 (Zhuang Zhao et al., 2013).

Por otra parte, el algoritmo genético también requiere de la definición de ciertos parámetros de funcionamiento como número de generaciones, probabilidades de cruce y mutación, entre otras. Asimismo, el modelo matemático implementado depende de constantes como el límite de consumo de energía c_u , el factor de ajuste del costo de la energía λ y demás variables tratadas en el capítulo 4.3. Para esto, se han programado los valores resumidos en la Tabla 4.9 (Zhuang Zhao et al., 2013).

6.1.1. Nuevo cronograma

Después de que se han registrado todas las variables necesarias para generar un nuevo cronograma, en la ventana *Mis cronogramas* se selecciona el botón *Crear nuevo cronograma* que se puede observar en el recuadro naranja de la Figura 5.12. El usuario será notificado sobre esta acción como se muestra en la Figura 6.2, sin embargo, la evolución del algoritmo sucederá detrás de la aplicación y de forma no visible para los usuarios como se puede observar en la Figura 6.1

```
[2019-06-02 17:09:00][25] Processing: App\Events\ScheduleRequested
[2019-06-02 17:09:00][25] Processed: App\Events\ScheduleRequested
[2019-06-02 17:09:00][26] Processing: App\Jobs\GenerateSchedule
Generation: 1(11.418407261179) Stall Generations: 0
Generation: 2(9.384923014011) Stall Generations: 0
Generation: 3(9.384923014011) Stall Generations: 1
Generation: 4(9.3640733898) Stall Generations: 0
Generation: 5(8.3695048873683) Stall Generations: 0
Generation: 6(7.3412057250038) Stall Generations: 0
Generation: 7(7.3412057250038) Stall Generations: 1
Generation: 8(6.3232971515063) Stall Generations: 0
Generation: 9(6.3200779321592) Stall Generations: 0
Generation: 10(6.3200779321592) Stall Generations: 1
```

Figura 6.1. Evolución de la generación del cronograma en el servidor.



Figura 6.2. Notificación de nuevo cronograma en proceso.

Durante el proceso de generación del nuevo cronograma, el usuario es libre de realizar cualquier acción dentro de la aplicación como verificar gráficas históricas de otros días, analizar variables en la base de datos, entre otras.

Una vez que el cronograma se ha generado, el sistema notifica al usuario tal y como indica la Figura 6.3. Ahora, la tabla de cronogramas generados muestra toda la información completa de los resultados obtenidos del algoritmo como generaciones evaluadas, valor fitness, el cromosoma y la fecha de generación.

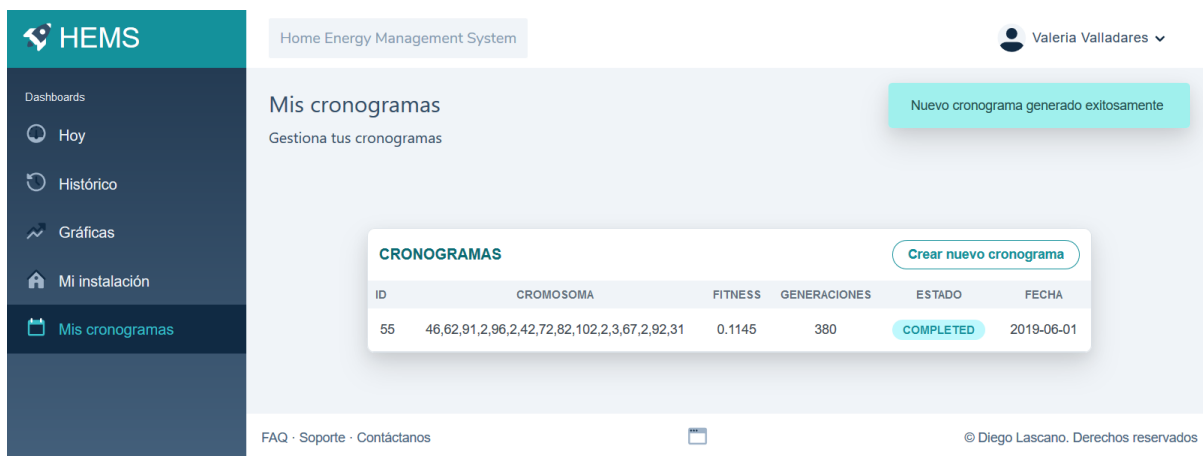


Figura 6.3. Generación de nuevo cronograma finalizado exitosamente.

En el lado del servidor también se puede observar el resultado mostrado en la anterior figura. El cromosoma se genera a través de un array de 16 elementos que corresponden a los 16 AOA's seleccionados para la generación del cronograma. En la Figura 6.4 se muestra la solución arrojada por el algoritmo genético después de haber iterado 380 veces con un valor fitness mínimo encontrado de 0.1145.

```

Generation: 377(0.1144667348716) Stall Generations: 98
Generation: 378(0.1144667348716) Stall Generations: 99
Generation: 379(0.1144667348716) Stall Generations: 100
App\Services\GeneticAlgorithm\Individual {#916
  -chromosome: array:16 [
    0 => 46
    1 => 62
    2 => 91
    3 => 2
    4 => 96
    5 => 2
    6 => 42
    7 => 72
    8 => 82
    9 => 102
    10 => 2
    11 => 3
    12 => 67
    13 => 2
    14 => 92
    15 => 31
  ]
  -fitness: 0.1144667348716
}
[2019-06-02 17:23:04][30] Processed: App\Jobs\GenerateSchedule

```

Figura 6.4. Generación del cronograma en el servidor finalizado exitosamente.

6.2. Análisis de resultados.

En primer lugar, es importante analizar el cronograma de funcionamiento óptimo generado por el algoritmo genético para ver si se cumplen las preferencias de usuario. Esto es, comparar el inicio del ciclo de operación propuesto por el cronograma con el ingresado como preferencia de usuario.

En la Tabla 6.1 se muestra una comparativa entre las preferencias de usuario y el inicio de funcionamiento propuesto por el cronograma de funcionamiento óptimo.

Tabla 6.1. Preferencias ingresadas por el usuario y propuestas por el cronograma óptimo.

AOA*	Preferencia de usuario			Cronograma	Comparativa	
	OTI* - Inicio	OTI* - Fin	LOT*	Inicio	Retardo	Estado
Aire acondicionado 1	41	60	5	46	5	OK
Aire acondicionado 2	61	85	5	62	1	OK
Aire acondicionado 3	86	120	10	91	5	OK
Radiador eléctrico 1	1	30	5	2	1	OK
Radiador eléctrico 2	91	115	10	96	5	OK
Olla arrocera 1	1	25	2	2	1	OK
Olla arrocera 2	41	60	2	42	1	OK
Olla arrocera 3	71	90	2	72	1	OK
Calentador de agua	81	105	3	82	1	OK
Lavaplatos	101	120	2	102	1	OK
Lavadora de ropa	1	60	5	2	1	OK
Tetera eléctrica 1	1	25	1	3	2	OK
Tetera eléctrica 2	66	85	1	67	1	OK
Humidificador 1	1	30	10	2	1	OK
Humidificador 2	91	120	10	92	1	OK
Secadora de ropa	30	40	5	31	1	OK

* ver el glosario de términos

Nota: valores expresados en timeslots. Cada timeslot = 12 min

Como se puede observar en la tabla anterior, el inicio de operación de todos los artefactos está dentro del rango preferido del usuario. Cabe resaltar que, en su mayoría, los artefactos iniciarían operación casi en el *timeslot* preferido del usuario, lo cual brinda un buen nivel de satisfacción del cliente.

Respecto a la interfaz de usuario, para visualizar los resultados que ofrece el cronograma generado, el usuario debe ingresar en la ventana denominada *Hoy*. En esta se muestran los resultados tanto a nivel de consumo de energía eléctrica como el coste que este genera.

Con el cronograma generado, en términos de coste se han obtenido los resultados que se muestran en la Figura 6.5. En esta ventana se despliegan 3 tarjetas de resumen diario que indican el costo de la energía del hogar con y sin energía fotovoltaica, además del ahorro económico del día. De igual manera, se presenta la evolución del costo del día mediante un gráfico.



Figura 6.5. Resultados del consumo energético del día en términos de coste.

Como se puede observar en la imagen, al final del día, utilizando el cronograma generado mediante el algoritmo genético, se ha conseguido un ahorro total de **34.06€**. Este resultado es obtenido considerando el aporte de energía fotovoltaica generada en la instalación que se está estudiando. Como se muestra en las tarjetas de resumen diario en la Figura 6.5, el costo generado exclusivamente por las cargas durante el día alcanza un valor de **1.54€**, sin embargo, el costo real obtenido es de **1.20€** lo cual indica el ahorro antes mencionado que alcanza un valor de **22.13%**.

Asimismo, en la gráfica *Tendencia de costo* de la Figura 6.5, se pueden observar los valores del costo de energía consumida a cada hora del día. Cabe indicar que en esta gráfica generada se muestran 2 grupos de datos; en color verde se presenta la evolución del costo de la energía consumida sin considerar le energía fotovoltaica mientras que en color rosa se incluye el aporte de energía generada. Se puede ver como se ha reducido

el costo de la energía consumida principalmente en las horas comprendidas entre las 8:00 y las 17:00, es decir, se evidencia el aporte de la energía fotovoltaica generada. En la Figura 6.6 se puede ver más a detalle el valor del costo para las 9:00. En esta se presenta una etiqueta que muestra un costo de **6.26¢/kWh** para los datos que corresponden al costo del consumo de energía sin considerar energía fotovoltaica generada, mientras que, en color rosa, este valor indica **0.00¢/kWh**.

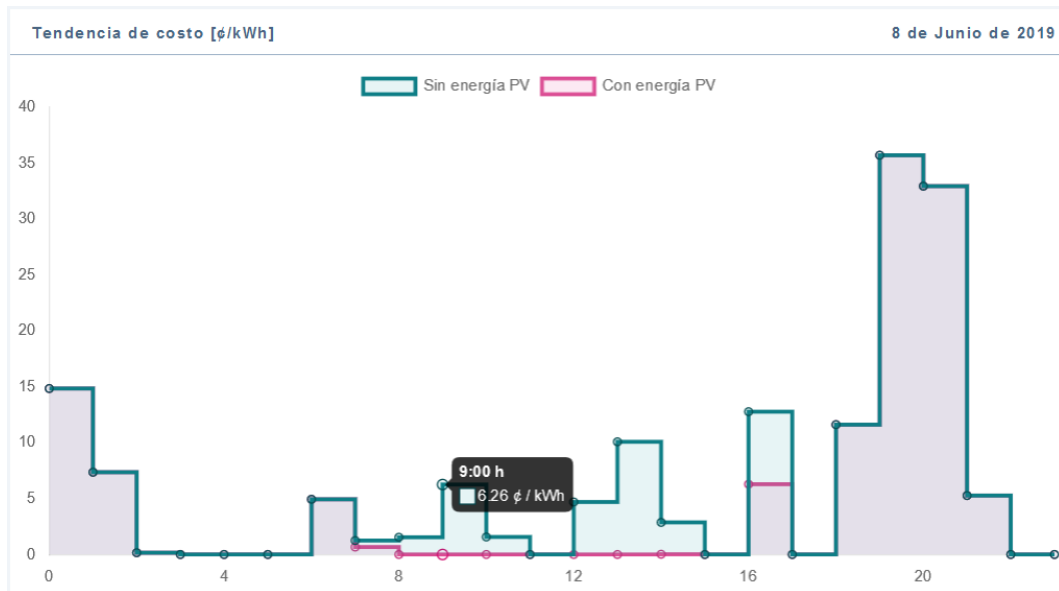


Figura 6.6. Tendencia de costo de la energía consumida para el día 8 de junio

Por otra parte, en términos de energía eléctrica se han obtenido los resultados que se muestran en la Figura 6.7. De igual manera que en la variable coste, se presentan 3 tarjetas de resumen diario con la energía consumida por todos los AOAs, la energía fotovoltaica generada y la energía ahorrada.

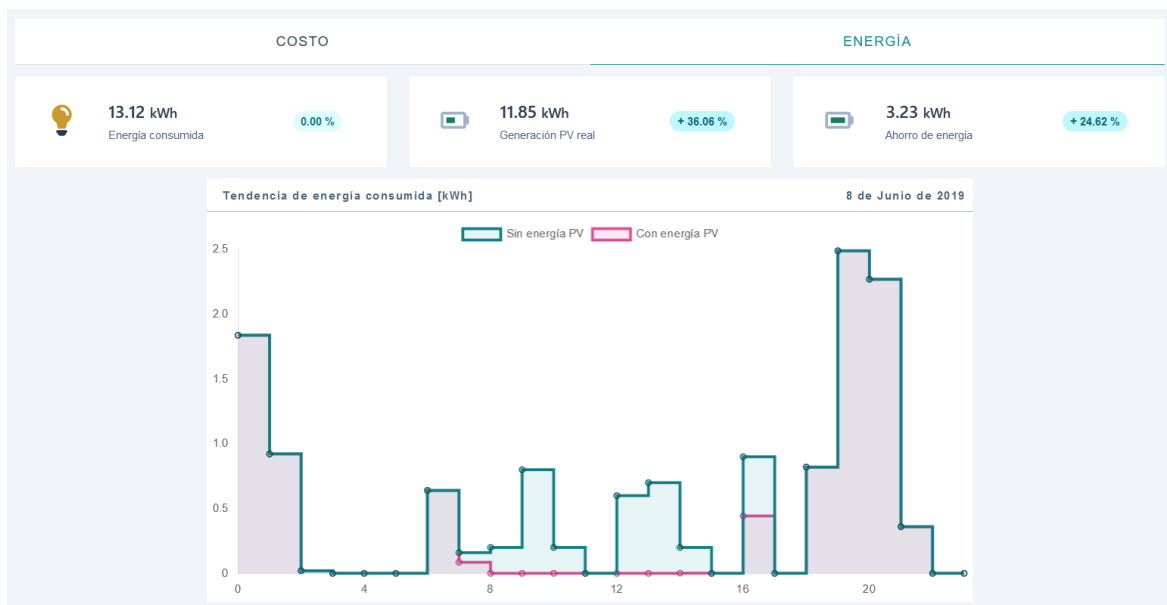


Figura 6.7. Resultados del consumo energético del día en términos de energía.

Con el cronograma generado, en términos de energía eléctrica se ha obtenido un consumo de $13.12kWh$ por parte de los AOAs. Además, el sistema de generación fotovoltaica diseñado indica una producción de $10.74kWh$, de los cuales nada más se han aprovechado $3.23kWh$ que representa un 24.62% de la energía total consumida; esto quiere decir que la mayoría de la energía fotovoltaica generada está siendo producida fuera del horario de mayor consumo de energía. El exceso de energía producida que no está siendo utilizada podría ser almacenada en un banco de baterías para un mayor aprovechamiento de la energía eléctrica producida.

La gráfica *Tendencia de energía consumida* por su parte, muestra la evolución del consumo energético a lo largo del día. Se muestran dos gráficas superpuestas para realizar una comparación visual. En color verde se muestra el consumo energético sin considerar el aporte de la energía fotovoltaica generada, mientras que en color rosa se muestra el mismo consumo incluyendo la energía fotovoltaica. En la Figura 6.8 se puede observar el dato de las 9:00h de forma puntual, donde en la línea de color verde que corresponde al consumo de energía sin considerar la energía fotovoltaica existe un consumo de energía de $0.8kWh$, mientras que en color rosa, al incluir la energía fotovoltaica, el valor de energía consumida de la red es de $0.00kWh$

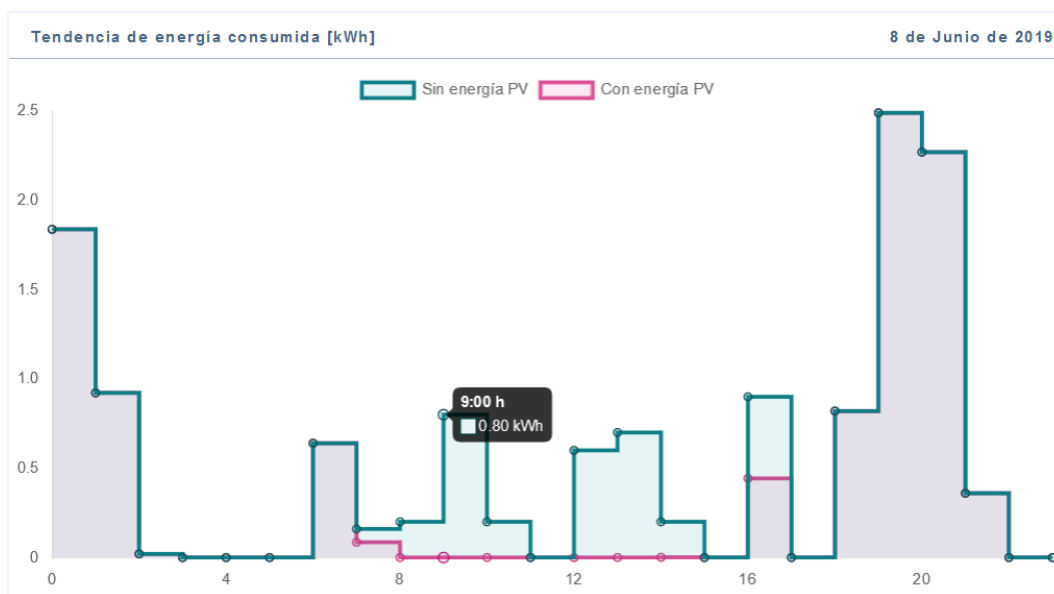


Figura 6.8. Tendencia de la energía consumida para el día 8 de junio.

7. CONCLUSIÓN

El objetivo principal de este proyecto fue diseñar y programar una aplicación multiplataforma de gestión de energía, capaz de generar cronogramas de funcionamiento para los artefactos automáticos del hogar, y así optimizar el uso de la instalación. Al final del proyecto realizado, se ha conseguido desarrollar esta aplicación denominada *Home Energy Management System*, considerando varios criterios de diseño importantes.

Se ha realizado el modelamiento matemático del consumo energético de un hogar, considerando principalmente artefactos de operación automática. Para esto se han definido dos aspectos fundamentales y sus prioridades. El primero es directamente el consumo de energía generado por los artefactos, y el segundo, la satisfacción del cliente que se interpreta como la rapidez con la que se cumple el ciclo de operación de cada artefacto.

Una de las actividades más importantes que se llevó a cabo fue el diseño e implementación del algoritmo genético. Este algoritmo evolutivo que ha sido programado en lenguaje PHP, es capaz de generar un cronograma de funcionamiento óptimo de los artefactos de operación automática del hogar. Se pudo comprobar que el cronograma generado cumple con las preferencias de funcionamiento ingresadas por el usuario.

Se ha conseguido implementar la aplicación HEMS mediante un *framework web* como es Laravel. De esta forma, cuando la aplicación sea llevada al estado de producción, será posible acceder a ella desde cualquier sistema operativo, y en cualquier dispositivo ya sea un computador de escritorio, o incluso dispositivos móviles.

En el Ecuador, el costo de la energía tiene un precio fijo para el sector residencial. Sin embargo, en el código fuente de la aplicación HEMS se ha implementado una estructura de costo dinámico adaptable a la realidad del país, de modo que, sin importar si el costo es fijo o si este se modifica varias veces al día, nada más es necesario alimentar la base de datos con la estructura correcta para que el algoritmo genético pueda generar un cronograma de funcionamiento óptimo, tal y como se ha realizado en el presente proyecto.

Como resultado indirecto del proyecto, se ha podido observar la importancia de incluir baterías en la instalación de generación fotovoltaica. En el caso particular de los datos de generación fotovoltaica del Ecuador obtenida mediante el software SAM, se pueden ver claramente altos niveles de energía generada, sin embargo, gran parte de esta energía (entre el 50% y el 75%) aparentemente no sería aprovechada por la instalación ya que está siendo producida fuera del horario de mayor consumo de energía.

8. PRESUPUESTO

El presupuesto estimado para el desarrollo del proyecto es el que se muestra en la Tabla 8.1

Tabla 8.1. Presupuesto general del proyecto.

HOME ENERGY MANAGEMENT SYSTEM					
Proyecto:		Desarrollo del sistema HEMS			
Elaborado por:	Diego Lascano				
Fecha:	04/06/2019				
Teléfono:	610053207				
Dirección:	Carrer Mare de Déu de la Fuencisla 78P				
Descripción					
El proyecto consiste en el diseño y desarrollo de una aplicación web para la gestión de energía eléctrica en los hogares. Este proyecto incluye la programación de algoritmos de inteligencia artificial para la generación de cronogramas de uso.					
ID	Descripción	Unidad	V. Unitario	Cantidad	V. Total
1	Diseño de la aplicación HEMS	horas	25,00 €	100	2.500,00 €
2	Programación de la aplicación HEMS	horas	30,00 €	250	7.500,00 €
3	Ordenador portátil	unidad	950,00 €	1	950,00 €
4	Software SAM	unidad	- €	1	- €
5	Framework Laravel	unidad	- €	1	- €
6	Matlab student como herramienta de apoyo	unidad	35,00 €	1	35,00 €
7	Artículos de investigación referentes al tema	unidad	15,00 €	8	120,00 €
				TOTAL	11.105,00 €

Este es un presupuesto referencial donde se indica el costo de todos los componentes inmiscuidos en el desarrollo del sistema HEMS.

9. TRABAJO FUTURO

Aunque se han cumplido los objetivos planteados al inicio del presente proyecto, durante el desarrollo de este han surgido varias mejoras que pueden ser aplicadas en trabajos futuros relacionados, los cuales son expuestos a continuación:

- Optimizar el algoritmo genético implementado para reducir el tiempo de cálculos de generación de cronogramas de optimización.
- Ampliar la información de energía fotovoltaica generada mediante simulación utilizando bases de datos de otras regiones, de modo que la aplicación HEMS pueda ser utilizada en otras zonas.
- Desarrollar/implementar hardware o un servicio que permita la lectura y registro del costo dinámico horario de la energía en tiempo real, con lo cual, los resultados generados por la aplicación HEMS serán más ajustados a la realidad.
- Implementar sensores de medición de variables ambientales tales como temperatura exterior, irradiancia, velocidad del viento, entre otras, para tener una base de datos de mediciones reales para los cálculos del cronograma de optimización.
- Modificar el código para que la aplicación realice múltiples actualizaciones al cronograma de optimización del día según las mediciones de variables externas.
- Desarrollar e implementar los protocolos de comunicación entre la aplicación HEMS con hardware específico para control de artefactos en el hogar, de modo que el cronograma generado pueda enviar órdenes a los artefactos operados automáticamente del hogar.
- Realizar un experimento para generar datos de consumo real de energía en un hogar, durante un periodo de tiempo más extenso, que permita evaluar la efectividad del algoritmo en una situación de la vida real, considerando una instalación sin cronograma de funcionamiento, y otra que funcione bajo este cronograma.

10. BIBLIOGRAFÍA

- Althaher, S., Mancarella, P., & Mutale, J. (2015). Automated Demand Response From Home Energy Management System Under Dynamic Pricing and Power and Comfort Constraints. *IEEE Transactions on Smart Grid*, 6(4), 1874-1883. <https://doi.org/10.1109/TSG.2014.2388357>
- Apple. (s. f.). macOS Mojave. Recuperado a partir de <https://itunes.apple.com/app/macOS-mojave/id1398502828?ls=1&mt=12>
- BP. (2018). 67 th edition Contents is one of the most widely respected. *Statistical Review of World Energy*, 1-56.
- CONELC, C. N. de E. (2013). Plan Maestro de Electrificación 2013 - 2022: Estudio y gestión de la demanda eléctrica. 1, 2, 1-206. Recuperado a partir de <https://www.regulacionelectrica.gob.ec/wp-content/uploads/downloads/2015/12/Vol2-Estudio-y-gestión-de-la-demanda-eléctrica.pdf>
- Corporation, M. (s. f.). Windows 10 - Microsoft Store España. Recuperado 11 de abril de 2019, a partir de <https://www.microsoft.com/es-es/store/b/windows>
- Espitia Cüchango, H. E., & Sofrony Esmeral, J. I. (2015). Algoritmo de optimización basado en enjambres de partículas con comportamiento de vorticidad y búsqueda individual y grupal. *Revista Tecnura*, 18(42), 24. <https://doi.org/10.14483/udistrital.jour.tecnura.2014.4.a02>
- Guayaquil, C. de I. de. (2017). Energía eléctrica y su incidencia en las actividades productivas, 1-7. Recuperado a partir de <http://www.industrias.ec/archivos/CIG/file/CARTELERA/2018/Competitividad-energía-electrica--.pdf>
- Haupt, R. L., & Haupt, S. H. (2004). *Practical Genetic Algorithm* (2.ª ed.). JOHN WILEY & SONS.
- Huang, Y., Tian, H., & Wang, L. (2015). Demand response for home energy management system. *International Journal of Electrical Power and Energy Systems*, 73, 448-455. <https://doi.org/10.1016/j.ijepes.2015.05.032>
- IDC. (2012). *IDC report into Server market share*. Idc.com. Recuperado a partir de <http://www.idc.com/about/viewpressrelease.jsp?containerId=prUS22360110§ionId=null&elementId=null&pageType=SYNOPSIS>
- Karaboga, D. (2005). *AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION* (Vol. 48).
- Laravel. (s. f.-a). Documentation - Laravel - The PHP Framework For Web Artisans. Recuperado 12 de abril de 2019, a partir de <https://laravel.com/docs/5.8>
- Laravel. (s. f.-b). Laravel Homestead - Laravel - The PHP Framework For Web Artisans. Recuperado 13 de mayo de 2019, a partir de <https://laravel.com/docs/5.8/homestead>

- MathWorks. (s. f.-a). *MATLAB Documentation*. MathWorks. Recuperado a partir de <http://www.mathworks.com/help/matlab/index.html>
- MathWorks. (s. f.-b). Pricing and Licensing - MATLAB & Simulink. Recuperado 30 de abril de 2019, a partir de <https://es.mathworks.com/pricing-licensing.html?prodcod=ML&intendeduse=comm>
- MathWorks. (2018). Company Overview. Recuperado a partir de <https://www.mathworks.com/content/dam/mathworks/tag-team/Objects/c/company-fact-sheet-8282v18.pdf>
- Microsoft. (s. f.). Precios y opciones de compra | Visual Studio. Recuperado 30 de abril de 2019, a partir de <https://visualstudio.microsoft.com/es/vs/pricing/>
- MSDN. (s. f.-a). Binding Controls to Data in Visual Studio. Recuperado 30 de abril de 2019, a partir de <https://msdn.microsoft.com/en-us/library/ms171923.aspx>
- MSDN. (s. f.-b). Breakpoint Overview. Recuperado 30 de abril de 2019, a partir de [https://msdn.microsoft.com/en-us/library/5557y8b4\(VS.80\).aspx](https://msdn.microsoft.com/en-us/library/5557y8b4(VS.80).aspx)
- National Instruments. (s. f.-a). 3rd Party Instrument Drivers - National Instruments. Recuperado 13 de abril de 2019, a partir de <http://www.ni.com/downloads/instrument-drivers/>
- National Instruments. (s. f.-b). Seleccione su Edición de LabVIEW - National Instruments. Recuperado 13 de abril de 2019, a partir de <http://www.ni.com/es-es/shop/labview/select-edition.html>
- National Instruments. (s. f.-c). Software Configuration Management and LabVIEW - National Instruments. *www.ni.com*. Recuperado a partir de <http://www.ni.com/white-paper/4114/en/>
- NREL. (s. f.). Home - System Advisor Model (SAM). Recuperado 7 de junio de 2019, a partir de <https://sam.nrel.gov/>
- Qiu, J., Shen, Y., Xie, J., & Wang, J. (2013). Gbest-guided artificial bee colony algorithm for global numerical function optimization. *International Journal of Applied Mathematics and Statistics*, 43(13), 117-125. <https://doi.org/10.1016/j.amc.2010.08.049>
- Red Eléctrica de España. (s. f.). Descargas | ESIOS electricidad · datos · transparencia. Recuperado 3 de mayo de 2019, a partir de https://www.esios.ree.es/es/descargas?date_type=datos&start_date=03-05-2019&end_date=03-05-2019
- Risteska Stojkoska, B. L., & Trivodaliev, K. V. (2017). A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 140, 1454-1464. <https://doi.org/10.1016/j.jclepro.2016.10.006>
- Selectra. (s. f.). ¿Qué significa tener una tarifa de acceso 2.0? Recuperado 3 de mayo de 2019, a partir de <https://tarifaluzhora.es/comparador/tarifas-electricas/2-0>
- Shareef, H., Ahmed, M. S., Mohamed, A., & Al Hassan, E. (2018). Review on Home

- Energy Management System Considering Demand Responses, Smart Technologies, and Intelligent Controllers. *IEEE Access*, 6, 24498-24509. <https://doi.org/10.1109/ACCESS.2018.2831917>
- Shen, J., Jiang, C., Liu, Y., & Wang, X. (2016). A Microgrid Energy Management System and Risk Management under an Electricity Market Environment. *IEEE Access*, 4, 2349-2356. <https://doi.org/10.1109/ACCESS.2016.2555926>
- Sivanandam, S. N., & Deppa, S. N. (2008). *Introduction to Genetic Algorithms*. New York: Springer Berlin Heidelberg.
- StatCounter. (s. f.). Desktop Operating System Market Share Worldwide | StatCounter Global Stats. Recuperado 11 de abril de 2019, a partir de <http://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-201803-201903-bar>
- Torres, M., & Baran, B. (2015). Optimización de enjambre de partículas para problemas de muchos objetivos. *Proceedings - 2015 41st Latin American Computing Conference, CLEI 2015*, 1-50. <https://doi.org/10.1109/CLEI.2015.7359982>
- Travis, J., & Kring, J. (2007). *LabVIEW for everyone : graphical programming made easy and fun*. (3rd ed.). Upper Saddle River, NJ: Prentice Hall.
- Venegas, M. (2018). *Proyecto de un sistema EMS de monitorización de una planta potabilizadora de agua, con paneles solares y baterías externas*. Terrassa.
- Vue.js. (s. f.). Introduction — Vue.js. Recuperado 13 de abril de 2019, a partir de <https://vuejs.org/v2/guide/>
- Zhang, Y., Zeng, P., Li, S., Zang, C., & Li, H. (2015). A Novel Multiobjective Optimization Algorithm for Home Energy Management System in Smart Grid. *Mathematical Problems in Engineering*, 2015, 1-19. <https://doi.org/10.1155/2015/807527>
- Zhuang Zhao, Won Cheol Lee, Yoan Shin, & Kyung-Bin Song. (2013). An Optimal Power Scheduling Method for Demand Response in Home Energy Management System. *IEEE Transactions on Smart Grid*, 4(3), 1391-1400. <https://doi.org/10.1109/tsg.2013.2251018>

ANEXOS

ANEXO A – Valor Técnico Ponderado

El valor técnico ponderado se define mediante la siguiente fórmula (Venegas, 2018):

$$VTP = \frac{\sum_{i=1}^n p_i * g_i}{p_{max} * \sum_{i=1}^n g_i}$$

Donde,

p_i : calificación de cada objeto.

g_i : valor del peso de cada criterio.

p_{max} : calificación máxima.

Al momento de dar una calificación a los criterios establecidos, se ha considerado una escala de 10 puntos, siendo 0 la calificación más baja y 10 la calificación más alta. Asimismo, la escala para los pesos (o nivel de importancia del criterio de selección) se ha fijado en 5 puntos, donde 0 es el valor más bajo y 5 el más elevado.

A continuación, se muestran los valores normalizados y resultados obtenidos aplicando el VTP para el **Sistema Operativo**

Criterio de selección	CALIFICACIÓN			Peso
	Windows	Linux	macOS	
Costo	0	10	10	5
Compatibilidad	9	7	4	4
Seguridad	6	8	8	4
Estabilidad	5	9	8	3
Soporte técnico	7	9	7	3

Criterio de selección	NORMALIZACIÓN			Peso
	Windows	Linux	macOS	
Costo	0	50	50	50
Compatibilidad	36	28	16	40
Seguridad	24	32	32	40
Estabilidad	15	27	24	30
Soporte técnico	21	27	21	30
VTP	0,51	0,86	0,75	190

A continuación, se muestran los valores normalizados y resultados obtenidos aplicando el VTP para el **Entorno de desarrollo**

Criterio de selección	CALIFICACIÓN				Peso
	Laravel	LabVIEW	MatLAB	Visual Studio	
Costo	10	5	5	10	5
Compatibilidad	10	7	10	9	4
Estabilidad	8	8	9	7	3
Soporte (comunidades online)	9	7	7	7	4
Capacitación requerida	5	6	5	6	5
Complejidad de uso	6	8	8	7	3

Criterio de selección	NORMALIZACIÓN				Peso
	Laravel	LabVIEW	MatLAB	Visual Studio	
Costo	50	25	25	50	50
Compatibilidad	40	28	40	36	40
Estabilidad	24	24	27	21	30
Soporte (comunidades online)	36	28	28	28	40
Capacitación requerida	25	30	25	30	50
Complejidad de uso	18	24	24	21	30
VTP	0,80	0,66	0,70	0,78	240

A continuación, se muestran los valores normalizados y resultados obtenidos aplicando el VTP para el **Algoritmo de optimización**

Criterio de selección	CALIFICACIÓN			Peso
	GA	PSO	ABC	
Conocimiento específico	7	7	7	4
Facilidad de implementación	9	7	7	5
Efectividad	10	8	8	4
Costo computacional	8	8	7	5
Fuentes de investigación	8	6	5	3

Criterio de selección	NORMALIZACIÓN			Peso
	GA	PSO	ABC	
Conocimiento específico	28	28	28	40
Facilidad de implementación	45	35	35	50
Efectividad	40	32	32	40
Costo computacional	40	40	35	50
Fuentes de investigación	24	18	15	30
VTP	0,84	0,73	0,69	210

ANEXO B – Hoja de datos del panel solar SPR-X21-335-BLK



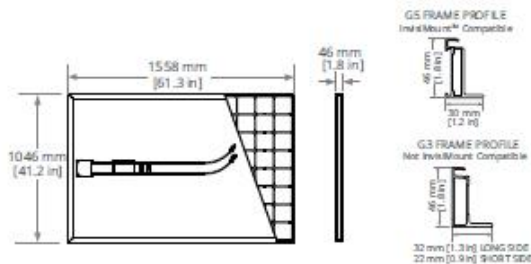
Electrical Data		
	SPR-X21-335-BLK	SPR-X21-345
Nominal Power (P _{nom}) ²	335 W	345 W
Power Tolerance	+5/-0%	+5/-0%
Avg. Panel Efficiency ⁴	21.0%	21.5%
Rated Voltage (V _{mpp})	57.3 V	57.3 V
Rated Current (I _{mpp})	5.85 A	6.02 A
Open-Circuit Voltage (V _{oc})	67.9 V	68.2 V
Short-Circuit Current (I _{sc})	6.23 A	6.39 A
Max. System Voltage	600 V UL & 1000 V IEC	
Maximum Series Fuse	15 A	
Power Temp Coef.	-0.29% / °C	
Voltage Temp Coef.	-167.4 mV / °C	
Current Temp Coef.	2.9 mA / °C	

Tests And Certifications	
Standard Tests ⁷	UL1703 (Type 2 Fire Rating), IEC 61215, IEC 61730
Management System Certs	ISO 9001:2015, ISO 14001:2015
EHS Compliance	RoHS, OHSAS 18001:2007, lead free, REACH SVHC-163, PV Cycle
Sustainability	Cradle to Cradle Certified™ Silver (contributes to LEED categories) ⁸
Ammonia Test	IEC 62716
Desert Test	10.1109/PVSC.2013.6744437
Salt Spray Test	IEC 61701 (maximum severity)
PID Test	1000V, IEC62804, PVEL 600hr duration
Available Listings	UL, TUV, MCS, CSA, FSEC, CEC

Operating Condition And Mechanical Data	
Temperature	-40° F to +185° F (-40° C to +85° C)
Impact Resistance	1 Inch (25 mm) diameter hail at 52 mph (23 m/s)
Appearance	Class A+
Solar Cells	96 Monocrystalline Maxeon Gen III
Tempered Glass	High-transmission tempered anti-reflective
Junction Box	IP-65, MC4 compatible
Weight	41 lbs (18.6 kg)
Max. Load	G5 Frame: Wind: 62 psf, 3000 Pa front & back Snow: 125 psf, 6000 Pa front
	G3 Frame: Wind: 50 psf, 2400 Pa front & back Snow: 112 psf, 5400 Pa front
Frame	Class 1 black anodized (highest AAMA rating)

REFERENCES:

- 1 SunPower 360W compared to a Conventional Panel on same sized arrays (260W, 16% efficient, approx. 1.6 m²), 4% more energy per watt (based on 3pty module characterization and PVSim), 0.75%/yr slower degradation (Campeau, Z. et al. "SunPower Module Degradation Rate," SunPower white paper, 2013).
- 2 "SunPower Module 40-Year Useful Life" SunPower white paper, May 2015. Useful life is 99 out of 100 panels operating at more than 70% of rated power.
- 3 X-Series same as E-Series, 5 of top 8 panel manufacturers tested in 2013 report, 3 additional panels in 2014. Ferrara, C., et al. "Fraunhofer PV Durability Initiative for Solar Modules: Part 2," Photovoltaics International, 2014.
- 4 See us.sunpower.com/home-solar-system-warranty/ for more details.
- 5 Standard Test Conditions (1000 W/m² irradiance, AM 1.5, 25° C, NREL calibration Standard, SOM5 current, IACCS FF and Voltage).
- 6 Based on average of measured power values during production.
- 7 Type 2 fire rating per UL1703:2013, Class C fire rating per UL1703:2002.
- 8 See salesperson for details.



G5 frames have no mounting holes. Please read the safety and installation guide.

Document # 504828 Rev G/LTR_US

©September 2017 SunPower Corporation. All rights reserved. SUNPOWER, the SUNPOWER logo, MAXEON, SIGNATURE and InvisiMount are trademarks or registered trademarks of SunPower Corporation. Specifications included in this datasheet are subject to change without notice.

SUNPOWER®

ANEXO C – System Advisor Model



System Advisor Model (SAM) General Description (Version 2017.9.5)

Nate Blair, Nicholas DiOrio, Janine Freeman,
Paul Gilman, Steven Janzou, Ty Neises,
and Michael Wagner
National Renewable Energy Laboratory

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy
Laboratory (NREL) at www.nrel.gov/publications.

Technical Report
NREL/TP-6A20-70414
May 2018

Contract No. DE-AC36-08GO28308

1 SAM Overview

This document describes the System Advisor Model (SAM) developed by the U.S. Department of Energy's National Renewable Energy Laboratory (NREL), and is written to help potential users determine whether the software meets their modeling needs, and to provide information for readers who do not plan to use SAM but want to learn about its capabilities.

SAM is a techno-economic computer model designed to facilitate decision making for people involved in the renewable energy industry:

- Project managers and engineers
- Financial and policy analysts
- Technology developers
- Researchers

SAM is available in the following platforms:

- Desktop application for Windows, Linux, and Mac OS
- Application programming interface (API) with a set of programming tools in the SAM software development kit (SDK)
- A set of documented open source C++ code repositories

To model a renewable energy project in SAM, you choose a performance model and a financial model to represent the project, and assign values to input variables to provide information about the project's location, type of equipment in the system, cost of installing and operating the system, and financial and incentives assumptions. Once you are satisfied with the input variable values, you run simulations, and then examine results. A typical analysis involves running simulations, examining results, revising inputs, and repeating that process until you understand and have confidence in the results.

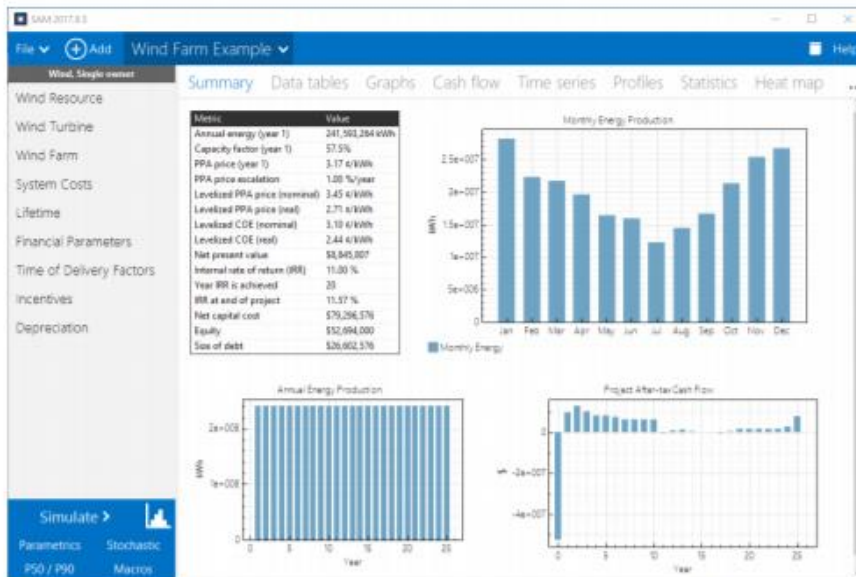


Figure 1. The SAM main window showing the results summary for a wind power system.

SAM's performance models are for photovoltaic systems with optional battery storage, concentrating solar power, industrial process heat, solar water heating, wind, geothermal, biomass, and conventional power systems that either deliver electricity directly to the power grid, or interact with the electric load of a grid-connected building or facility. SAM does not model off-grid power systems, or hybrid power systems with more than one power generation source. The financial models are for projects that either buy and sell electricity at retail rates (residential and commercial), or sell electricity at a price determined in a power purchase agreement (PPA). SAM can model large or small projects, ranging from residential rooftop photovoltaic installations to large concentrating solar power generation projects and wind farms.

SAM is an open source project, so its source code is available to the public. Researchers can study the code to understand the model algorithms, and software programmers can contribute their own models and enhancements to the project. Reference manuals describing the model algorithms are also available for download from the SAM website.

2 Software Downloads

NREL distributes SAM for free at <https://sam.nrel.gov>. It is available in the three packages for different applications as described below.

2.1 Desktop Application

SAM is available as a desktop application for Windows, Mac, and Linux computers. The current version at the time of this writing is Version 2017.9.5, released in September 2017. The desktop version of SAM is a complete application that provides a graphical user interface to set values of inputs, configure and run simulations, and generate tables and graphs of results. The SAM desktop application is available at <https://sam.nrel.gov/download>.

NREL releases one or two desktop versions of SAM each year, and makes available legacy versions on the SAM website.

2.2 Software Development Kit

The SAM Software Development Kit (SDK) is a package of tools for software developers to create applications that interact with the SAM Simulation Core (SSC). It provides access to SSC via the same application programming interface (API) that the SAM desktop application uses. The SAM SDK is available at <https://sam.nrel.gov/sdk>.

The SDK contains:

- The SSC API, `sccapi.h`
- The SSC dynamic library and supporting libraries for Windows, Mac, and Linux
- The SDKtool application
- The SSC Guide
- Code examples



Figure 2. The Generate Code command on the Case menu exports ready-to-run code from the desktop application to use with the software development kit.

The desktop application's Code Generator can generate ready-to-run code in any of the following languages from the inputs in a SAM case:

- C
- MATLAB
- Python 2 and 3
- Java

3 Inputs and Default Values

SAM requires input data to describe the performance characteristics of physical equipment in the system, and project costs and financial assumptions. The desktop application comes with default input values and tools for downloading some inputs from online data services. For SDK users, when you run SSC via the API, you must assign values to the inputs in your code. (You can use the code generator described in Section 2.2 above.)

SAM requires a weather data file as input to describe the renewable energy resource and weather conditions at a project location. In the desktop application, you either choose a weather data file from a list, download one from the internet, or create the file using your own data.

The desktop application comes with several libraries of performance data and coefficients that describe the characteristics of commercially available system components such as photovoltaic modules and inverters, parabolic trough receivers and collectors, wind turbines, and biopower combustion systems. For those components, you simply choose an option from a list, and SAM applies values from the library to the input variables.

The desktop application comes with a set of component libraries that store input parameters for the photovoltaic, solar water heating, wind and parabolic trough performance models:

- Module parameters from the California Energy Commission
- Inverter parameters from the California Energy Commission
- Solar hot water collector parameters from the SRCC
- Trough receiver parameters from NREL
- Trough collector parameters from NREL
- Wind turbine power curves from NREL

The desktop application can also automatically download data and populate input variable values from the following online databases:

- [OpenEI Utility Rate Database](#) for retail electricity rate structures for U.S. utilities.
- [NREL National Solar Radiation Database](#) for solar resource data and ambient weather conditions.
- [NREL Wind Integration Datasets](#) for wind resource data.
- [NREL Biofuels Atlas](#) and [DOE Billion Ton Update](#) for biomass resource data.

For the remaining input variables, you either use the default value or change its value. Some examples of input variables are:

- Installation costs including equipment purchases, labor, engineering and other project costs, land costs, and operation and maintenance costs.
- Numbers of modules and inverters, tracking type, and derating factors for photovoltaic systems.
- Collector and receiver type, solar multiple, storage capacity, and power block capacity for parabolic trough systems.
- Analysis period, real discount rate, inflation rate, tax rates, internal rate of return target or power purchase price for utility financing models.
- Building load and time-of-use retail rates for commercial and residential financing models.
- Tax and cash incentive amounts and rates.