



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

**PROGRAMACIÓN DE UNA GUI EN
MATLAB PARA EL CÁLCULO DE
INSTALACIONES SOLARES
TÉRMICAS PARA ACS Y
CALEFACCIÓN**

Alumno: Manuel Gonzalo Alcázar Vargas

Tutor: Prof. D. David Vera Candeas

Depto.: Ingeniería Eléctrica

Septiembre, 2018

To be a good professional engineer, always start to study late for exams, because it teaches you how to manage time and tackle emergencies

Bill Gates

Índice general

Índice de figuras	VII
Índice de tablas	IX
1. Introducción	1
1.1. Energía solar térmica	1
1.2. Partes y tipología de instalaciones solares térmicas	6
1.3. Revisión de los métodos matemáticos de cálculo para instalaciones solares térmicas de baja temperatura	10
2. Materiales y métodos	11
2.1. Desarrollo matemático del método seleccionado	11
2.2. Programación e implementación en MATLAB del método de cálculo seleccionado	23
2.3. Elaboración de una herramienta de cálculo virtual a través de una GUI en MATLAB intuitiva para el cálculo de instalaciones solares tér- micas de BT	26
3. Análisis de los resultados	29
3.1. Aplicación de la herramienta de cálculo en diferentes escenarios . . .	29
3.2. Validación y análisis de los resultados	36
4. Conclusiones finales y propuesta de mejora	39
Anexos	42
A. PVGIS	45
B. Código GUI.m	51
Bibliografía	71

Índice de figuras

1.1. Horno solar usado por Lavoisier en 1774. Fuente: [1]	3
1.2. Colector parabólico suministrando energía a una imprenta en la exposición de París de 1878. Fuente: [1]	4
1.3. Esquemas de una instalación de circuito abierto (izquierda) y circuito cerrado (derecha). Fuente: [12]	7
1.4. Esquemas de una instalación de circulación natural (izquierda) y circulación forzada (derecha). Fuente: [12]	7
1.5. Configuraciones tipo <i>termosifón</i> . Fuente: [1]	8
1.6. Esquema básico de un sistema solar térmico de baja temperatura para ACS. Fuente [4]	9
2.1. Tipologías de sistemas para los que es adecuado el método <i>f-chart</i> . Fuente: [1]	13
2.2. Zonas climáticas. Fuente: [16]	16
2.3. Vista de la GUI desde el entorno de MATLAB «guide»	27
2.4. Vista de la GUI desde el entorno de MATLAB de usuario	28
2.5. Opciones para abrir una GUI desde el entorno <i>guide</i>	28
3.1. Ventana de inicio de la GUI	30
3.2. Ventana de trabajo de la GUI	30
3.3. Paso 1: Provincia y municipio	31
3.4. Paso 2: Tipo de edificio y número de unidades de consumo	31
3.5. Paso 3: Calcular	32
3.6. Paso 4: Recalcular	33
3.7. Caso 2: Vivienda unifamiliar en Sevilla	34
3.8. Caso 3: Polideportivo en Jaén. Excesiva superficie de captación	35
3.9. Caso 3: Polideportivo en Jaén. Superficie de captación adecuada	35
3.10. Caso 4: Residencia de ancianos en Tudela (Navarra)	36
3.11. Ortofotografía de la Real Casa de la Misericordia en Tudela (Navarra). Fuente: Google Maps	37
A.1. Radiación mensual PVGIS	46
A.2. Configuración para descargar la radiación solar	46
A.3. Archivo de texto generado por la página PVGIS	47

Índice de tablas

1.1. Sistemas solares de calentamiento de agua	9
2.1. Rango en el que se mueven las variables de diseño para el desarrollo del método <i>f-chart</i> , tanto para sistemas líquidos como de aire. Fuente: [7]	12
2.2. Demanda de referencia a 60°C	14
2.3. Número de personas en función del número de dormitorios de una vivienda	15
2.4. Radiación solar global	15
2.5. Contribución solar mínima en %. Caso general	17
2.6. Contribución solar mínima en %. Caso Efecto Joule	17
2.7. Contribución solar mínima en %. Caso climatización en piscinas . . .	17
2.8. Temperatura diaria media mensual de agua fría (°C)	19
2.9. Temperatura diaria media mensual de agua fría (°C) (continuación) . .	20
3.1. Comparativa entre los resultados procedentes de la aplicación e instalaciones solares ya existentes. Fuente elaboración propia y [12] . .	36

Capítulo 1

Introducción

1.1. Energía solar térmica

La energía solar es la forma de energía más antigua que se ha utilizado. El sol era adorado por la práctica totalidad de las civilizaciones antiguas y lo veneraban como a un dios poderoso. La primera aplicación práctica conocida del sol se encuentra en el secado de la comida para preservarla [28].

Probablemente la aplicación a gran escala más antigua conocida hasta la fecha sea la destrucción, prendiendo en llamas, de la flota romana en la bahía de Siracusa por Arquímedes. Los científicos han discutido largo y tendido sobre este tema durante siglos. Desde el s. I A.C. hasta el s. XI D.C. autores han hecho referencia a este evento; aunque posteriormente se desmintió pues la tecnología del momento no estaba suficientemente desarrollada como para construir espejos [19]. La cuestión era si Arquímedes tenía el conocimiento suficiente de óptica como para diseñar un instrumento para concentrar la luz del sol en un punto a partir del cual las naves pudiesen empezar a arder desde la distancia. En cualquier caso, a Arquímedes se le atribuye la autoría de un libro, *On burning mirrors*, [21] que tan solo es conocido por referencias, ya que ninguna copia ha sobrevivido hasta nuestros días.

El historiador griego Plutarco, se refería al incidente diciendo que los romanos, viendo ese enorme poder que no sabían de dónde venía, se sintieron sobrecogidos y pensaron que estaban combatiendo contra los dioses.

En su libro, *Optics Vitelio*, un matemático polaco describió la quema de la flota romana en detalle [20], [19]: «El espejo de fuego de Arquímedes se componía de

24 espejos que convergían los rayos del sol en un punto y producían un exceso de calor.»

Porclus repitió el experimento de Arquímedes durante el periodo bizantino y quemó la flota de los enemigos asediando Bizancio en Constantinopla [19].

Dieciocho siglos después de Arquímedes, Atanasio Kircher realizó una serie de experimentos para prender desde la distancia una hoguera y así comprobar la validez científica de la historia de Arquímedes, pero no queda ningún informe de sus experimentos [21].

Muchos historiadores, por el contrario, creen que Arquímedes no utilizó espejos sino los escudos de los soldados, configurando un gran parábola, concentrando los rayos del sol en un punto de las naves. Este hecho probó que la radiación solar podía ser una gran fuente de energía. Muchos siglos después, los científicos reconsideraron la radiación solar como fuente de energía, intentando convertirla en una forma de energía directamente utilizable.

Las primeras aplicaciones de la energía solar se refieren al uso de colectores solares. Estos son, por su naturaleza, construcciones de geometría compleja y los requerimientos para seguir al sol son complicados. Durante el siglo XVIII, los hornos solares eran capaces de fundir hierro, cobre y otros metales. Estos hornos se construían de hierro pulido, lentes de vidrio y espejos. Los hornos se utilizaban en Europa y Oriente Medio. Una de las primeras aplicaciones a gran escala fue el horno solar construido por el famoso químico Lavoisier quien, en torno a 1774 construyó lentes potentes para concentrar la radiación solar (Figura 1.1). Este horno alcanzaba la temperatura de 1.750°C . El horno se componía de una lente de 1,32 m y una lente secundaria de 0,2 m para obtener esta temperatura, que resultó ser la máxima alcanzada durante 100 años. Otra aplicación de la energía solar en el mismo siglo la realizó el naturalista francés Boufon, que experimentó con diferentes artilugios que describió como «espejos calientes quemando a gran distancia» [20].

Durante el siglo XIX, se realizaron intentos de convertir energía solar en otras formas de energía basándose en la generación de vapor de baja presión para operar motores de vapor. Augusto Monchot se podría considerar el padre de este campo al construir y operar diversos motores de vapor movidos por energía solar entre los años 1864 y 1878 en Europa y el norte de África. Uno de ellos se presentó

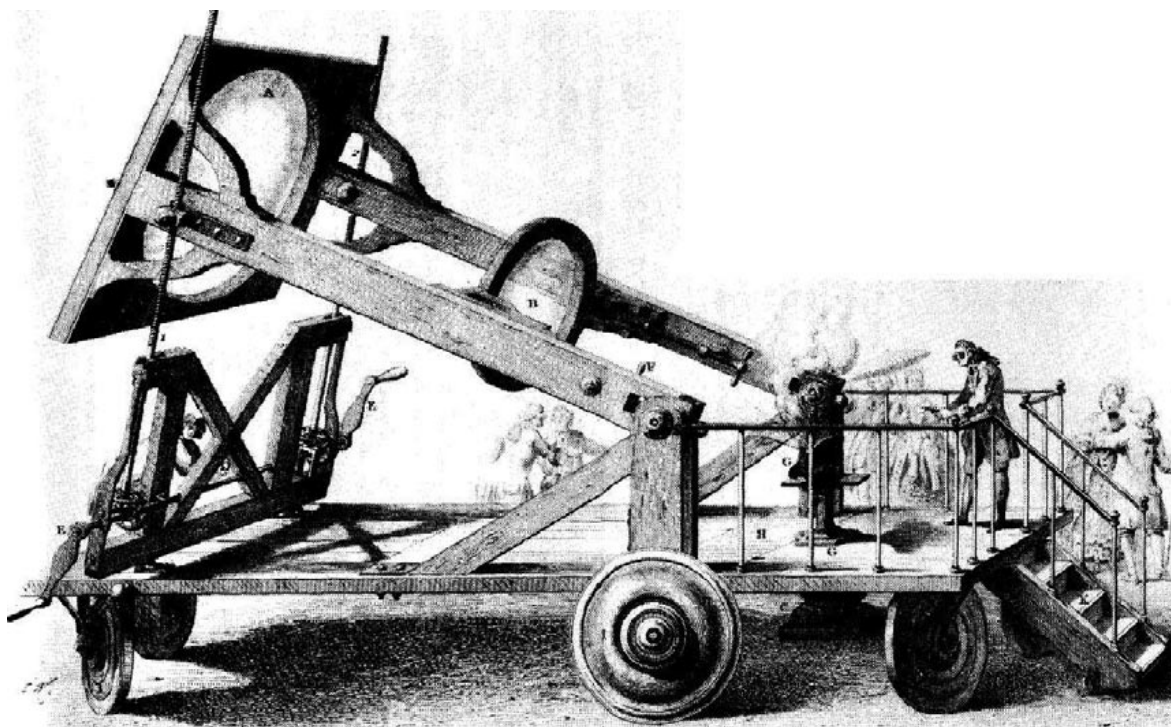


Figura 1.1: Horno solar usado por Lavoisier en 1774. Fuente: [1]

en la exposición de París de 1878 (Figura 1.2). La energía solar se utilizaba para producir el vapor que suministraba energía a una imprenta [22], [23]. La evaluación de uno construido en Tours por el gobierno francés demostró que era económicamente inviable. También se construyó otro en Argelia. En 1875, Mouchot realizó un gran avance en el diseño de colectores solares al construir uno con forma de cono truncado. El colector de Mouchot consistía en una placa compuesta de pequeñas láminas de plata y con un diámetro de 5,4 m y un área de colector de 18,6 m². Las partes móviles tenían una masa de 1.400 kg.

Abel Pifre, contemporáneo a Mouchot, también construyó motores solares [21], [24]. Los colectores solares de Pifre eran reflectores parabólicos compuestos de pequeños espejos. Aparentemente eran similares a los conos truncados de Mouchot.

Se siguió investigando en Estados Unidos, donde el ingeniero americano John Ericsson desarrolló el primer motor de vapor que funcionaba directamente con energía solar. Ericsson construyó ocho sistemas parabólicos que empleaban aire o agua como fluido de trabajo [26].

En el año 1901, A. G. Eneas instaló un colector de 10 m de diámetro que suministraba energía a una bomba de agua en una granja de California. El aparato constaba de dos estructuras tipo paraguas abierto e invertido y dispuesto en un

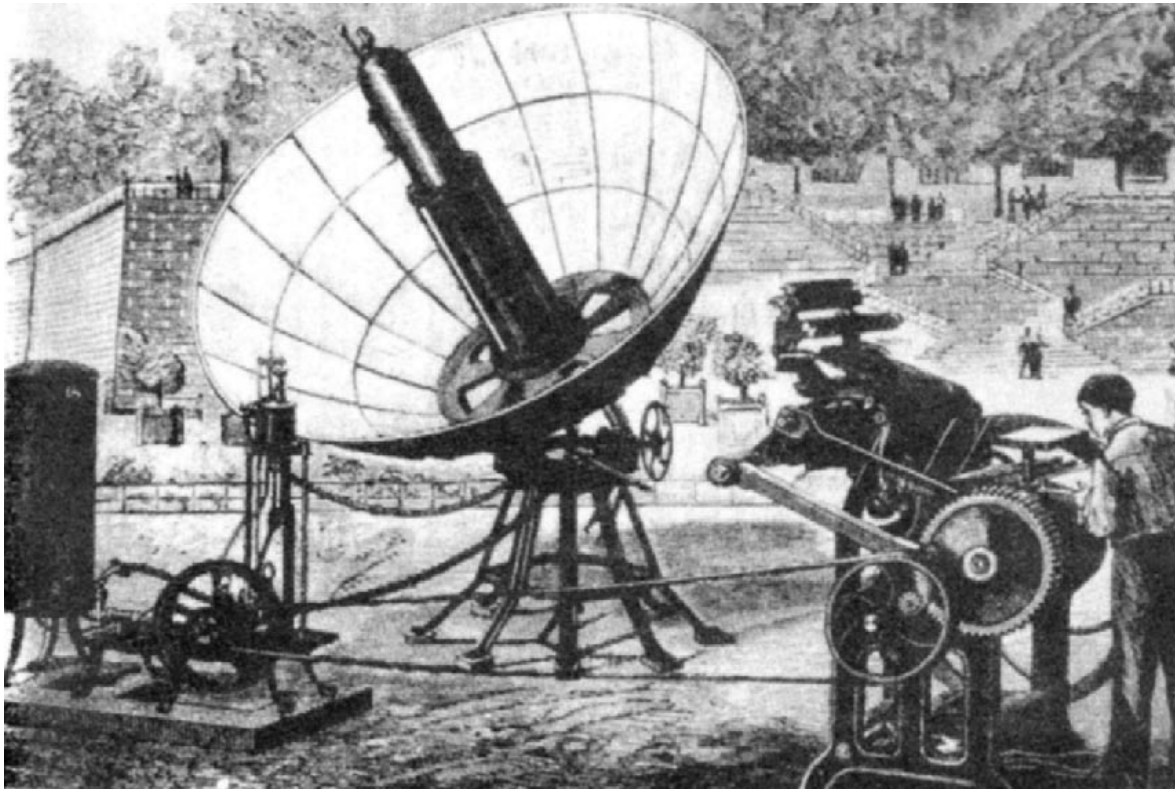


Figura 1.2: Colector parabólico suministrando energía a una imprenta en la exposición de París de 1878. Fuente: [1]

ángulo para recibir los rayos del sol en los 1.788 espejos que conformaban la superficie reflectora. Los rayos del sol eran concentrados en un punto donde estaba la caldera. El agua dentro de la caldera se calentaba y se convertía en vapor para producir energía mecánica que accionaba la bomba centrífuga [25].

En 1904, un fraile portugués, Padre Himalaya, construyó un gran horno solar. Éste se expuso en la feria mundial de St. Louis. El horno aparentemente era una estructura moderna, grande y con una especie de cuerno parabólico que hacía de colector [21].

En 1912, Frank Shuman, en colaboración con C. V. Boys, se dispusieron a construir la central de bombeo más grande del mundo, en Maadi (Egipto). El sistema entró en funcionamiento en 1913, utilizando largos cilindros parabólicos para concentrar la luz del sol en un tubo. Cada cilindro tenía 62 m de largo y el área colectora total era de 1.200 m². La planta desarrollaba una potencia total de 37 a 45 kW de modo continuo, durante 5 h. A pesar del éxito de la planta, fue cerrada en el año 1915 debido a la primera guerra mundial y a la bajada del precio del petróleo.

Durante los siguientes 50 años, se realizaron múltiples variaciones de los dise-

ños comentados anteriormente, siempre con el fin de calentar un fluido de trabajo para conseguir potencia mecánica. Las dos tecnologías principales utilizadas fueron receptores centrales y receptores distribuidos empleando varios puntos de concentración o líneas para concentrar la radiación solar. Los sistemas de recepción central emplean campos de heliostatos (sistemas de seguimiento solar de dos ejes) para concentrar la radiación solar en una única torre instalada como receptor central [27]. Los sistemas de recepción distribuidos emplean discos parabólicos, lentes de *Fresnel* y cilindros parabólicos. Los discos parabólicos siguen al sol en dos ejes y utilizan espejos para concentrar la radiación en un único punto receptor. Los cilindros parabólicos son reflectores que concentran la radiación en unos receptores tubulares a lo largo de las líneas focales. El rango de temperaturas de los receptores varía desde 100°C en sistemas de baja temperatura hasta cerca de 1.500°C en sistemas de recepción central [27].

Hoy en día, muchas plantas solares tienen una potencia del orden de megavatios para producir energía eléctrica o calor. La primera planta solar comercialmente instalada se construyó en Albuquerque, Nuevo México, en 1979. Se componía de 220 heliostatos y tenía una potencia de 5 MW. La segunda se instaló en Barstow, California, con una potencia térmica total de 35 MW. La mayoría de las plantas solares producen electricidad o calor para procesos industriales y suministran vapor recalentado a temperaturas del orden de 400°C. Por tanto, pueden producir electricidad o vapor para hacer funcionar pequeñas plantas desalinizadoras tanto por energía térmica o eléctrica.

Otra área de interés, especialmente para el presente trabajo fin de grado, es la relacionada con el agua caliente sanitaria y la calefacción, la cual apareció en la década de 1930, pero no ganó popularidad hasta la segunda mitad de los años 40. Hasta entonces, millones de hogares se calentaban con calderas de carbón. La idea es calentar el agua y que alimente al sistema de radiadores que ya está instalado en la vivienda.

La fabricación de colectores solares para agua caliente empezó en los años 60. La industria de colectores solares se expandió muy rápidamente por multitud de países del mundo.

1.2. Partes y tipología de instalaciones solares térmicas

Quizás la aplicación más conocida de los sistemas solares térmicos sea el calentamiento de agua con fines domésticos. La popularidad de estos sistemas se basa en el hecho de que son relativamente simples y que el calentamiento de agua con fines domésticos suele ser viable en la mayoría de las ocasiones. Este tipo de instalaciones solares se denominan *instalaciones solares térmicas de baja temperatura*.

El consumo de energía térmica de baja temperatura se estima en torno a 10 EJ (10×10^{18} J) o 3×10^{11} kWh para agua caliente, lo que necesitaría en torno a 6 billones de m² de área de colectores [29]. En el año 2005, estaban instalados alrededor de 140 millones de m² de área de colectores, que es tan solo el 2,3% el potencial [30].

Un calentador solar es una combinación de un *array* de colectores solares, un sistema de transferencia de energía y un acumulador. La parte principal del calentador solar es el *array* de colectores solares, que absorbe la radiación solar y la convierte en calor. Posteriormente, este calor es absorbido por un fluido de transferencia de calor (normalmente agua glicolada o algún otro líquido con propiedades anticongelantes) que pasa a través del colector. El calor puede ser almacenado o utilizado directamente. Debido a que partes de la instalación solar están expuestas a las condiciones meteorológicas, éstas deben encontrarse debidamente protegidas frente a heladas y sobrecalentamientos debidos a periodos de alto aislamiento térmico y baja demanda energética. Existen dos tipos de sistemas de calentadores solares, a saber:

- Sistemas directos (o de circuito abierto), en los que el agua potable es calentada directamente en el colector, y
- Sistemas indirectos(o de circuito cerrado), en los que el agua potable es calentada indirectamente a través de un intercambiador de calor mediante el fluido de trabajo que es el que se calienta en el captador.

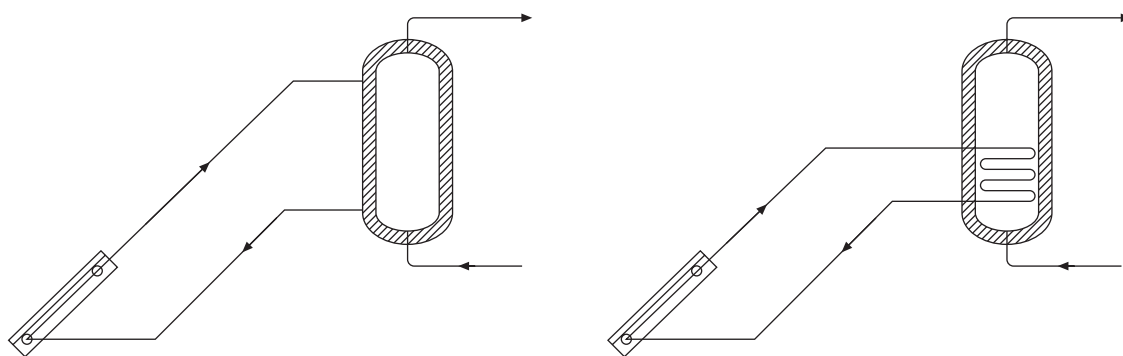


Figura 1.3: Esquemas de una instalación de circuito abierto (izquierda) y circuito cerrado (derecha). Fuente: [12]

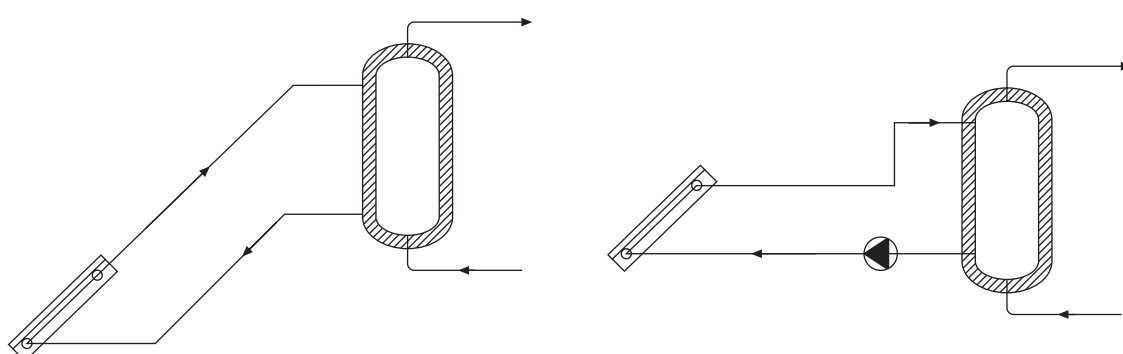


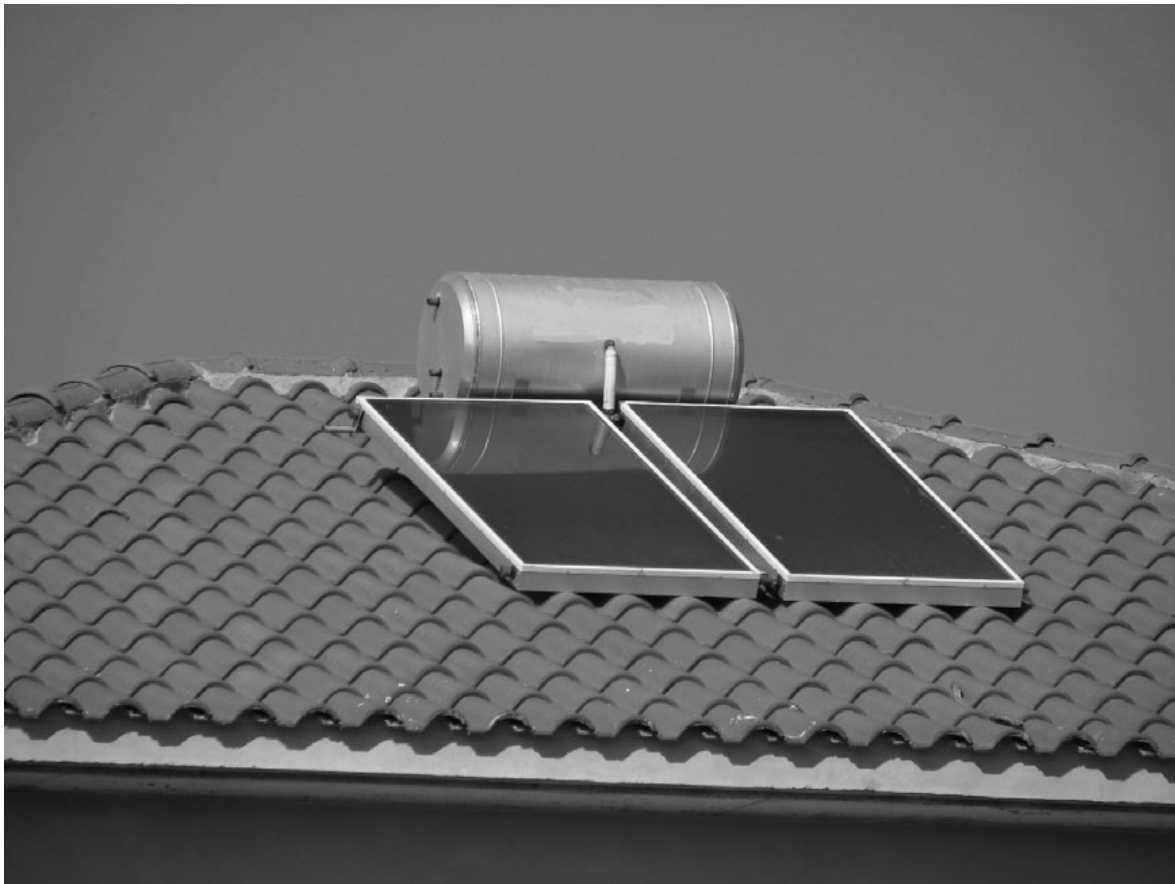
Figura 1.4: Esquemas de una instalación de circulación natural (izquierda) y circulación forzada (derecha). Fuente: [12]

Además, los sistemas se pueden clasificar según la manera en la que el fluido de trabajo es movido, a saber:

- Sistemas naturales (o pasivos)
- Sistemas de circulación forzada (o activos)

La circulación natural ocurre debido a la convección natural (termosifón), mientras que la circulación forzada precisa de bombas o ventiladores para mover el fluido de trabajo a través del colector. Exceptuando el termosifón y el sistema de colector con acumulador integrado, que no necesitan de ningún tipo de control, los sistemas de ACS de baja temperatura (BT) se controlan utilizando termostatos diferenciales.

Diversos tipos de sistemas de energía solar térmica se pueden utilizar para ACS y calefacción, tal y como se muestra en la tabla 1.1. El termosifón y el sistema de colector con acumulador integrado se denominan *sistemas pasivos*, puesto que



(a) Colector solar plano



(b) Tubo de vacío

Figura 1.5: Configuraciones tipo *termosifón*. Fuente: [1]

Tabla 1.1: Sistemas solares de calentamiento de agua

Sistemas pasivos	Sistemas activos
Termosifón (directo e indirecto) Colector con acumulador integrado	Sistemas de circulación directa Sistemas de circulación indirecta con intercambiadores de calor internos y externos Sistemas de aire Sistemas de bomba de calor Sistemas de calentamiento de piscinas

no precisan de ninguna bomba para su funcionamiento; mientras que los demás se denominan *sistemas activos*, ya que precisan de una bomba para su correcto funcionamiento. Para proteger de las heladas se emplean sistemas de recirculación y de desagüe.

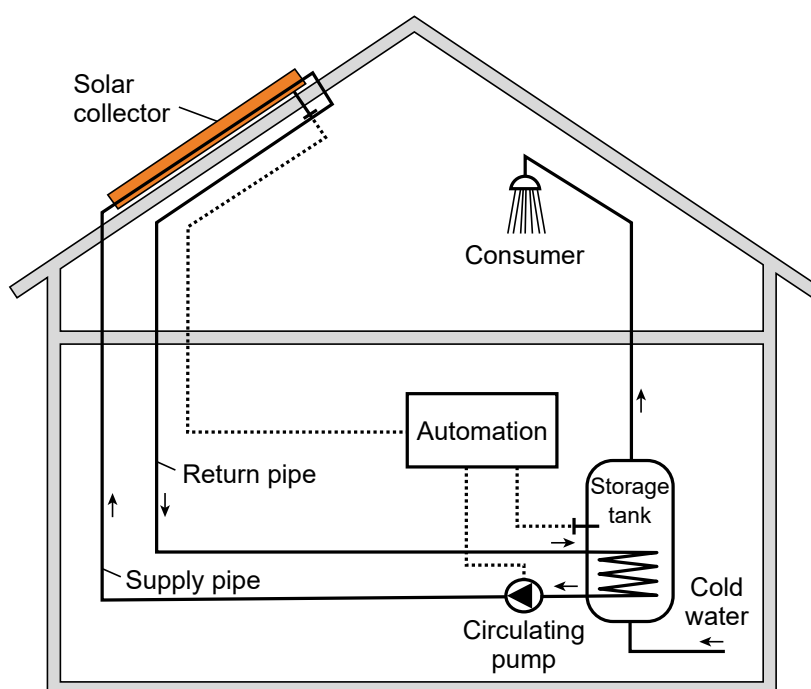


Figura 1.6: Esquema básico de un sistema solar térmico de baja temperatura para ACS. Fuente: [4]

Se han empleado diversos tipos de colectores para aplicaciones de calentamiento de agua, como son el colector plano, el colector de vacío, y colectores compuestos parabólicos. Además de estos tipos, para sistemas de mayor envergadura se pueden emplear tipologías más complejas como son los cilindros parabólicos.

La cantidad de agua caliente producida por un sistema de calentadores solares

depende del tipo y el tamaño del sistema, de la radiación solar disponible y de la demanda térmica del momento.

1.3. Revisión de los métodos matemáticos de cálculo para instalaciones solares térmicas de baja temperatura

El correcto dimensionamiento de los componentes de un sistema solar térmico es un problema complejo, que incluye, por un lado características predecibles como son el funcionamiento de los colectores y demás componentes; y por otro lado datos meteorológicos, cuya incertidumbre es elevada. Se va a presentar una introducción a los métodos de cálculo y programas actualmente utilizados para el cálculo de sistemas de energía solar térmica de baja temperatura.

Los métodos de diseño que se van a exponer son: el método *f-chart*, el método de **utilización** Φ y el método Φ, f -*chart*. El método *f-chart* se basa en la correlación de los resultados de un gran número de simulaciones en términos de variables adimensionales fácilmente calculables. El método de *utilización* Φ (o simplemente método de utilización) se utiliza en casos de que la temperatura de operación del colector se conoce o se puede estimar, y para los que los niveles críticos de radiación son conocidos. Este método se basa en el análisis horario de la radiación para obtener la fracción de la radiación solar mensual que está por encima del nivel crítico. El método Φf -*chart* es una combinación del método de utilización y del *f-chart*., aplicados a sistemas en los que la energía aportada a la carga se encuentra por encima de la temperatura mínima de uso y la temperatura de esta fuente de energía no tiene efecto en el funcionamiento del sistema siempre y cuando supere la temperatura mínima.

Para resultados más precisos, son necesarias simulaciones y modelos más precisos. En los últimos años, debido al aumento de la potencia computacional, las simulaciones anuales están empezando a tomar más importancia que los métodos de diseño anteriormente comentados. En cualquier casos, estos métodos son mucho más rápidos y, por tanto, todavía se encuentran en uso, así como en estu-

dios previos. Como paquetes de software para el cálculo de instalaciones solares térmicas se pueden citar TRNSYS, WATSUN y Polysun, entre otros. Además, actualmente se emplean técnicas de inteligencia artificial aplicadas a la predicción y modelado de sistemas solares térmicos.

Capítulo 2

Materiales y métodos

2.1. Desarrollo matemático del método seleccionado

El método del *f-chart* se utiliza para estimar el funcionamiento térmico anual de un sistema de calentamiento activo en un edificio empleando un fluido de trabajo, que bien puede ser líquido o gaseoso, y en el que la temperatura mínima de suministro está próxima a 20°C. Las configuraciones que pueden ser evaluadas mediante el método del *f-chart* son habitualmente edificios residenciales; aunque también se pueden evaluar otro tipo de edificaciones siempre que no haya un gran retraso entre la generación de ACS y el consumo de la misma. Con el método del *f-chart*, se estima la fracción de la carga térmica total aportada por el sistema solar. Si se denomina Q_a a la energía total necesaria y Q_u a la energía total aportada por el sistema solar, para el mes i -ésimo, la fracción en la que se reduce la energía que es necesario aportar mediante energía comprada cuando se emplea un sistema solar se denomina **fracción solar**, f , y viene definida por:

$$f = \frac{Q_a - Q_{\text{comprada}}}{Q_a} = \frac{Q_u}{Q_a} \quad (2.1)$$

El método del *f-chart* fue desarrollado por [6], [7] y [8]. En este método, la principal variable de diseño es el área de los colectores, mientras que otras variables, como son el volumen de acumulación, el tipo de colector, la carga térmica, el tamaño del intercambiador de calor o el caudal de fluido de trabajo toman un papel secundario. El método es una correlación de los resultados de cientos de simulaciones de

sistemas solares térmicos, realizados con TRNSYS, en el que las condiciones de la simulación variaban en todo el rango que se consideraba posible en un sistema práctico (tabla 2.1). Los resultados de las correlaciones proporcionan f , es decir, la fracción de la carga térmica mensual suministrada como energía solar térmica, como función de dos parámetros adimensionales. El primero está relacionado con la ratio entre el calor perdido en los colectores y la carga térmica, mientras que el segundo está relacionado con el calor absorbido en los colectores y la carga térmica. Estos parámetros, en la literatura anglosajona se suelen denotar mediante las letras X e Y respectivamente, mientras que en el caso de la literatura española [11] se emplean las variables D_2 y D_1 . En el presente trabajo fin de grado se empleará esta última notación.

Dentro del método del f -chart existen tres modelos según sea el sistema que se estudie, a saber: sistemas líquidos para calefacción, sistemas de aire para calefacción y sistemas para agua caliente sanitaria (Figura 2.1). La ecuación que permite obtener la fracción solar en el caso de los sistemas líquidos (tanto para ACS como para calefacción) es la (2.2); mientras que en el caso de que el fluido de trabajo sea aire, será la ecuación (2.3):

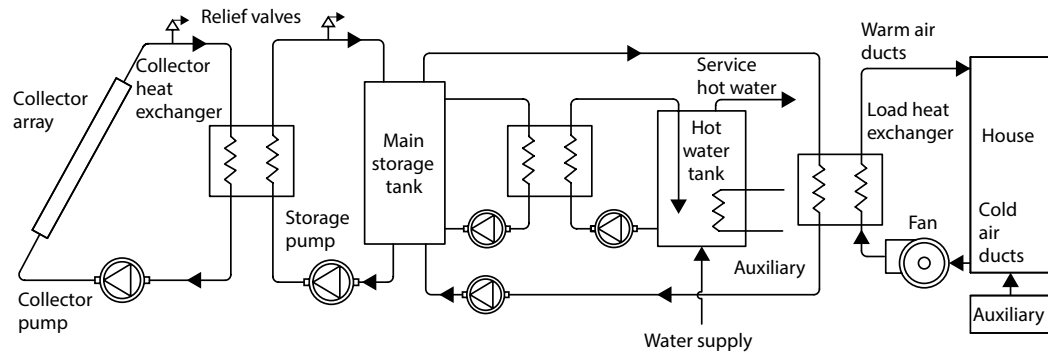
$$f = 1,029D_1 - 0,065D_2 - 0,245D_1^2 + 0,0018D_2^2 + 0,0215D_1^3 \quad (2.2)$$

$$f = 1,040D_1 - 0,065D_2 - 0,159D_1^2 + 0,00187D_2^2 + 0,0095D_1^3 \quad (2.3)$$

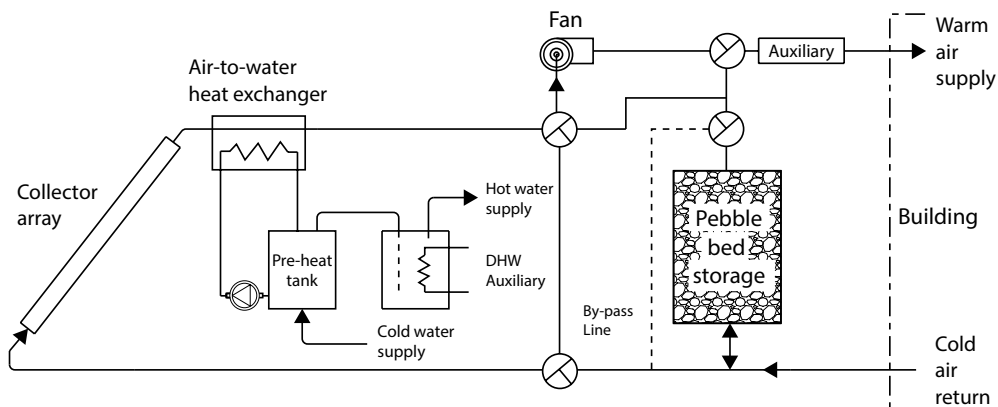
Tabla 2.1: Rango en el que se mueven las variables de diseño para el desarrollo del método f -chart, tanto para sistemas líquidos como de aire. Fuente: [7]

Parámetro	Rango
$(\tau\alpha)_n$	0,6 - 0,9
$F'_R S_c$	5 - 120 m ²
U_L	2,1 - 8,3 W/m ² °C
β (pte. colector)	30 - 90°
$(UA)_h$	83,3 - 666,6 W/°C

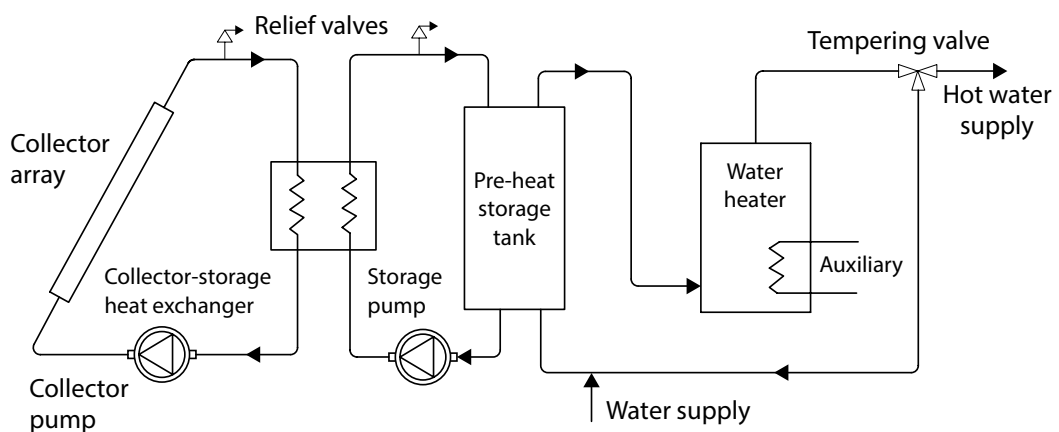
A continuación, se va a describir el proceso que es necesario seguir para determinar los parámetros D_1 y D_2 según el Instituto para la Diversificación y el Ahorro de Energía (IDAE) [11]:



(a) Sistema líquido para calefacción



(b) Sistema de aire para calefacción



(c) Sistema para agua caliente sanitaria

Figura 2.1: Tipologías de sistemas para los que es adecuado el método *f-chart*.
Fuente: [1]

1. Valoración de las cargas térmicas para ACS o calefacción (salidas)
2. Valoración de la radiación solar incidente sobre la superficie de captadores (entradas)
3. Cálculo del parámetro D_1
4. Cálculo del parámetro D_2
5. Determinación de los valores f
6. Cálculo de la cobertura solar mensual
7. Valoración de la cobertura solar anual

El primer paso en el cálculo de la instalación será determinar la demanda de referencia a 60°C. Para ello, se emplea la tabla 3.1 de [16], que a continuación se reescribe:

Tabla 2.2: Demanda de referencia a 60°C. Fuente: [16]

Criterio de demanda	Litros de ACS/día a 60°C	
Hospitales y clínicas	55	camas
Hotel ****	70	camas
Hotel ***	55	camas
Hotel/Hostal **	40	camas
Camping	40	emplazamientos
Hostal / Pensión *	35	camas
Residencia	55	camas
Vestuarios/Duchas colectivas	15	servicios/día
Escuelas sin ducha	3	alumnos
Cuarteles	20	personas
Fábricas y talleres	15	personas
Oficinas	3	personas
Gimnasios	25	usuarios/día
Restaurantes	10	comidas/día
Cafeterías	1	almuerzos/día
Vivienda unifamiliar	30	persona
Vivienda multifamiliar	22	persona

En caso de que la temperatura de acumulación sea diferente de 60°C, denotándose T a la temperatura de acumulación y T_i a la temperatura media del agua fría en el mes, se deberá emplear las expresiones (2.4) y (2.5) para el cálculo de la nueva demanda térmica. Nótese que en caso de que $T = 60^\circ\text{C}$, $D(T) = D(60^\circ\text{C})$.

$$D_i(T) = D_i(60^\circ\text{C}) \times \frac{60 - T_i}{T - T_i} \quad (2.4)$$

$$D(T) = \sum_{i=1}^{12} D_i(T) \quad (2.5)$$

En el caso de viviendas, tanto unifamiliares como multifamiliares, será necesario determinar el número de personas. En caso de que no se conozca, se empleará la siguiente tabla que relaciona el número de dormitorios con el número de personas:

Tabla 2.3: Número de personas en función del número de dormitorios de una vivienda

Nº dormitorios	1	2	3	4	5	6	7	>7
Nº personas	1,5	3	4	6	7	8	9	Nº de dormitorios

Por otro lado, y aunque se encuentra fuera del cálculo de la demanda, es preciso destacar el hecho de que **la demanda de agua en los hoteles de 4 estrellas es el doble que la demanda correspondiente a los hostales** (Tabla 2.2). Esto hace plantearse si la eficiencia energética debe centrarse principalmente en disminuir las emisiones de CO₂ o si por el contrario debería centrar sus esfuerzos en reducir el consumo.

El segundo paso para el cálculo de la instalación será determinar la contribución solar mínima que corresponde al municipio en el que se ubique la instalación. Para ello, existen dos métodos. El primero, empleando el mapa de las zonas climáticas (Figura 2.2) y el segundo a partir de la radiación solar global media diaria anual sobre superficie horizontal (H) (Tabla 2.4). En el presente TFG se empleará el último método pues el PVGIS proporciona el valor de H para cada municipio. Para los municipios más importantes y las capitales de provincia, se puede emplear la tabla 3.3 de [16].

Una vez determinada la zona climática a la que corresponde el municipio, será necesario determinar la contribución solar mínima, según las tablas 2.1, 2.2 y 2.3 de [16] que aquí se reescriben por mera comodidad (Tablas 2.5, 2.6 y 2.7). El CTE considera *caso general* aquel en el que la energía térmica que no es aportada por energía solar procede de combustibles fósiles (caldera de gas natural, gasóleo, fuel-oil, etc...) mientras que el caso de *efecto Joule* es aquel en el que la fuente

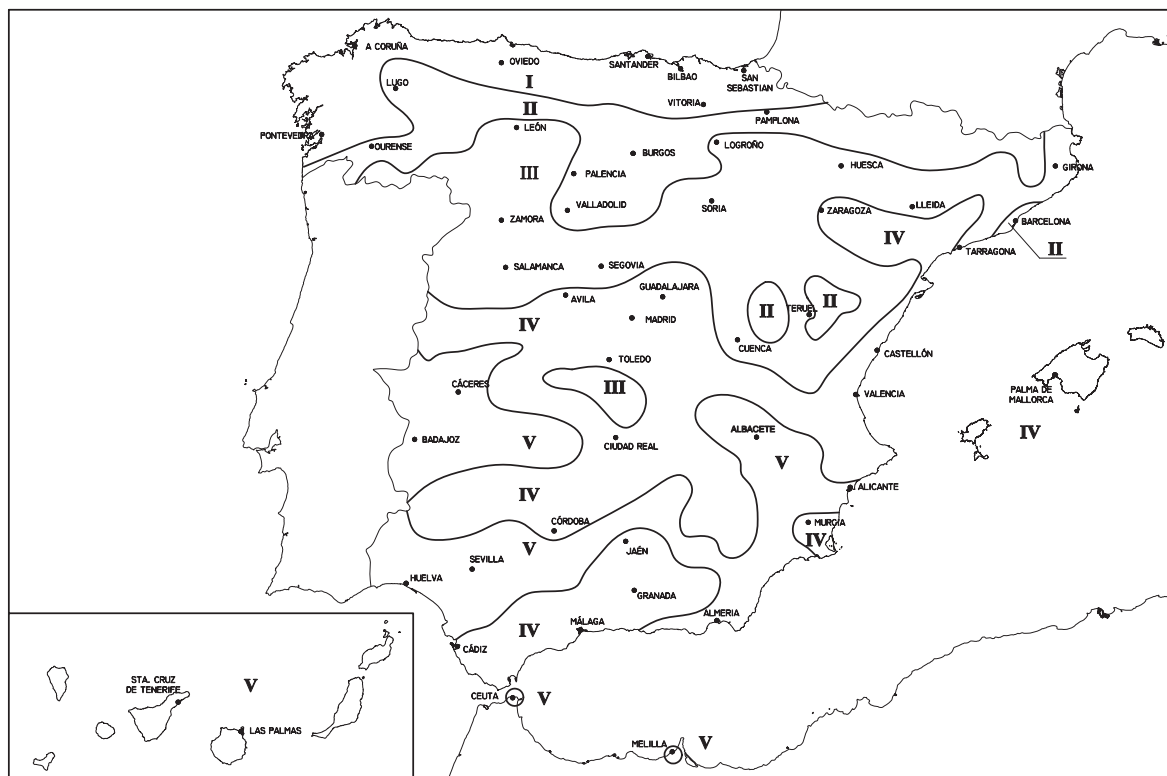


Figura 2.2: Zonas climáticas. Fuente: [16]

Tabla 2.4: Radiación solar global

Zona climática	kWh/m ²
I	$H < 3,8$
II	$3,8 \leq H < 4,2$
III	$4,2 \leq H < 4,6$
IV	$4,6 \leq H < 5,0$
V	$5,0 \leq H$

auxiliar de energía es una resistencia eléctrica. Es importante notar la enorme discrepancia que hay entre ambas contribuciones, **penalizando muy negativamente los calentadores eléctricos**.

Para la **valoración de las cargas térmicas**, es necesario, por un lado, conocer la demanda (en litros/día) de ACS. A continuación, será necesario aplicar la ecuación para determinar la cantidad de energía necesaria cada día:

$$Q_{ACS}[\text{kJ/día}] = D(T)[\text{L/día}] \cdot \rho_{\text{agua}}[\text{kg/L}] \cdot c_p[\text{kJ/(kg} \cdot ^\circ\text{C)}] \cdot (T_{ACS} - T_{\text{red}})[^\circ\text{C}] \quad (2.6)$$

Para determinar la temperatura del agua de la red (T_{red}) se emplearán las tablas 2.8 y 2.9. En caso de que se quiera calcular la temperatura de la red en un municipio

Tabla 2.5: Contribución solar mínima en %. Caso general

Demanda total de ACS del edificio (L/día)	Zona climática				
	I	II	III	IV	V
<5.000	30	30	50	60	70
5.000 - 6.000	30	30	55	65	70
6.000 - 7.000	30	35	61	70	70
7.000 - 8.000	30	45	63	70	70
8.000 - 9.000	30	52	65	70	70
9.000 - 10.000	30	55	70	70	70
10.000 - 12.500	30	65	70	70	70
12.500 - 15.000	30	70	70	70	70
15.000 - 17.500	35	70	70	70	70
17.500 - 20.000	45	70	70	70	70
>20.000	52	70	70	70	70

Tabla 2.6: Contribución solar mínima en %. Caso Efecto Joule

Demanda total de ACS del edificio (L/día)	Zona climática				
	I	II	III	IV	V
<1.000	50	60	70	70	70
1.000 - 2.000	50	63	70	70	70
2.000 - 3.000	50	66	70	70	70
3.000 - 4.000	51	69	70	70	70
4.000 - 5.000	58	70	70	70	70
5.000 - 6.000	62	70	70	70	70
>6.000	70	70	70	70	70

Tabla 2.7: Contribución solar mínima en %. Caso climatización en piscinas

	Zona climática				
	I	II	III	IV	V
Piscinas cubiertas	30	30	50	60	70

que no es la capital de provincia (no tabulado), se empleará la expresión:

$$T_{\text{red,municipio}} = T_{\text{red,capital provincia}} - B \cdot (Z_{\text{municipio}} - Z_{\text{capital provincia}}) \quad (2.7)$$

Donde:

- $B = 0,010$ para los meses de octubre a marzo y,
- $B = 0,005$ para los meses de abril a septiembre

Una vez se tiene la demanda térmica diaria (kWh/día), se multiplica por el número de días de cada mes para determinar la demanda térmica mensual, Q_a , (kWh/mes)

Por otro lado, debido a la naturaleza de la mayoría de las instalaciones solares térmicas para ACS, el ciclo de trabajo es diario. Esto es, la energía térmica entra al sistema durante las horas diarias y se gasta a lo largo del día, de modo que no se almacena energía de un día para otro. Partiendo de esta premisa, el volumen de acumulación será aproximadamente igual a la demanda térmica. De hecho, el IDAE [11] propone los siguientes límites:

$$0,8 < \frac{V_{\text{acumulación}}}{D(T)} < 1,2 \quad (2.8)$$

Como primera estimación, se tomará $V_{\text{acumulación}} = D(T)$. Además, según [8], el funcionamiento anual de este tipo de sistemas es insensible a la capacidad de almacenamiento del mismo, siempre y cuando se cumpla la relación:

$$\frac{V_{\text{acumulación}}}{S_c} > 50[\text{L/m}^2] \quad (2.9)$$

Siendo habitual emplear como estimación inicial de superficie de colectores el valor:

$$S_c[\text{m}^2] = \frac{V_{\text{acumulación}}[\text{L}]}{75} \quad (2.10)$$

Una vez se tiene una superficie inicial de captación ($S_{c,0}$), lo siguiente será determinar el número de colectores necesarios para conseguir esa superficie de captación. Es decir, si se obtiene una superficie de captación de $12,3 \text{ m}^2$ y cada colector tiene una superficie de captación de 2 m^2 , serán necesarios seis colectores y

Tabla 2.8: Temperatura diaria media mensual de agua fría (°C)

Provincia	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Santiago de Compostela	10	10	11	12	13	14	16	16	15	14	12	11
Albacete	7	8	9	11	14	17	19	19	17	13	9	7
Alicante/Alacant	11	12	13	14	16	18	20	20	19	16	13	12
Almería	12	12	13	14	16	18	20	21	19	17	14	12
Ávila	6	6	7	9	11	14	17	16	14	11	8	6
Badajoz	9	10	11	13	15	18	20	20	18	15	12	9
Barcelona	9	10	11	12	14	17	19	19	17	15	12	10
Bilbao	9	10	10	11	13	15	17	17	16	14	11	10
Burgos	5	6	7	9	11	13	16	16	14	11	7	6
Cáceres	9	10	11	12	14	18	21	20	19	15	11	9
Cádiz	12	12	13	14	16	18	19	20	19	17	14	12
Castellón/Castelló	10	11	12	13	15	18	19	20	18	16	12	11
Ceuta	11	11	12	13	14	16	18	18	17	15	13	12
Ciudad Real	7	8	10	11	14	17	20	20	17	13	10	7
Córdoba	10	11	12	14	16	19	21	21	19	16	12	10
Cuenca	6	7	8	10	13	16	18	18	16	12	9	7
Donostia-San Sebastián	9	9	10	11	12	14	16	16	15	14	11	9
Girona	8	9	10	11	14	16	19	18	17	14	10	9
Granada	8	9	10	12	14	17	20	19	17	14	11	8
Guadalajara	7	8	9	11	14	17	19	19	16	13	9	7
Huelva	12	12	13	14	16	18	20	20	19	17	14	12
Huesca	7	8	10	11	14	16	19	18	17	13	9	7
Jaén	9	10	11	13	16	19	21	21	19	15	12	9
León	6	6	8	9	12	14	16	16	15	11	8	6
Lleida	7	9	10	12	15	17	20	19	17	14	10	7

Tabla 2.9: Temperatura diaria media mensual de agua fría (°C) (continuación)

Provincia	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
Logroño	7	8	10	11	13	16	18	18	16	13	10	8
Lugo	7	8	9	10	11	13	15	15	14	12	9	8
Madrid	8	8	10	12	14	17	20	19	17	13	10	8
Málaga	12	12	13	14	16	18	20	20	19	16	14	12
Melilla	12	13	13	14	16	18	20	20	19	17	14	13
Murcia	11	11	12	13	15	17	19	20	18	16	13	11
Ourense	8	10	11	12	14	16	18	18	17	13	11	9
Oviedo	9	9	10	10	12	14	15	16	15	13	10	9
Palencia	6	7	8	10	12	15	17	17	15	12	9	6
Palma	11	11	12	13	15	18	20	20	19	17	14	12
Pamplona/Iruña	7	8	9	10	12	15	17	17	16	13	9	7
Salamanca	6	7	8	10	12	15	17	17	15	12	8	6
Santander	10	10	11	11	13	15	16	16	16	14	12	10
Segovia	6	7	8	10	12	15	18	18	15	12	8	6
Sevilla	11	11	13	14	16	19	21	21	20	16	13	11
Soria	5	6	7	9	11	14	17	16	14	11	8	6
Tarragona	10	11	12	14	16	18	20	20	19	16	12	11
Teruel	6	7	8	10	12	15	18	17	15	12	8	6
Toledo	8	9	11	12	15	18	21	20	18	14	11	8
Valencia/València	10	11	12	13	15	17	19	20	18	16	13	11
Valladolid	6	8	9	10	12	15	18	18	16	12	9	7
Vigo	10	11	11	13	14	16	17	17	16	14	12	10
Vitoria-Gasteiz	7	7	8	10	12	14	16	16	14	12	8	7
Zamora	6	8	9	10	13	16	18	18	16	12	9	7
Zaragoza	8	9	10	12	15	17	20	19	17	14	10	8

$S_c = S_{c,unitaria} \cdot n_{colectores} = 12 \text{ m}^2$. Con esta superficie de captación, S_c , se procede a calcular la fracción solar, f , por el método f -chart. Para ello, se comienza por determinar el parámetro D_1 . Según explica el IDAE, [11], indica la relación entre la energía absorbida, E_a y la carga calorífica total durante un mes, Q_a . La energía absorbida se calcula mediante la expresión:

$$E_a = S_c F'_r(\tau\alpha) R_1 N \quad (2.11)$$

Donde:

- S_c es la superficie total de captación [m^2]
- R_1 es la radiación diaria media mensual sobre la superficie de captación por unidad de área [kJ/m^2]
- $F'_r(\tau\alpha)$ es un factor adimensional, que viene dado por la ecuación:

$$F'_r(\tau\alpha) = F_r(\tau\alpha)_n \frac{(\tau\alpha)}{(\tau\alpha)_n} \frac{F'_r}{F_r} \quad (2.12)$$

Donde:

- $F_r(\tau\alpha)_n$ es rendimiento óptico del captador, también representado por η_o
- $(\tau\alpha)/(\tau\alpha)_n$ es un modificador del ángulo de incidencia. Según la superficie del captador sea sencilla o doble, se tomará el valor de 0,96 o 0,94 respectivamente
- F'_r/F_r es un factor de corrección del conjunto captador-intercambiador. Se suele tomar el valor de 0,95

Una vez determinado el parámetro D_1 , se calcula el parámetro que relaciona las pérdidas de energía en el captador con la carga calorífica total durante un mes, D_2 . La expresión para el cálculo de la energía perdida en el captador es:

$$E_p = S_c F'_r U_L (100 - t_a) \Delta t K_1 K_2 \quad (2.13)$$

Donde:

- S_c es la superficie total de captación [m^2]

- $F'_r U_L = F_r U_L (F'_r / F_r)$

Donde:

$F_r U_L$ es la pendiente de la curva característica del captador (coeficiente global de pérdidas del captador), a veces denominado k_1

- t_a es la temperatura media mensual del ambiente durante las horas diurnas
- Δt es el periodo de tiempo considerado, en segundos ($86.400 \text{ s/día} \cdot N_{\text{días}}$)
- K_1 es el factor de corrección por almacenamiento, que viene dado por la expresión:

$$K_1 = \left(\frac{\text{kg de acumulación}}{75 \cdot S_c} \right)^{-0,25} \quad (2.14)$$

- K_2 es el factor de corrección para ACS, que relaciona la temperatura mínima de ACS con la media mensual del ambiente. Viene dada por la expresión:

$$K_2 = \frac{11,6 + 1,18t_{ac} + 3,86t_r - 2,32t_a}{100 - t_a} \quad (2.15)$$

Donde:

- t_{ac} : temperatura mínima requerida del ACS
- t_r : temperatura del agua de la red
- t_a : temperatura media mensual del ambiente *en las horas diurnas*

Una vez determinados E_a y E_p se emplea las expresiones:

$$D_1 = \frac{E_a}{Q_a} \quad (2.16)$$

$$D_2 = \frac{E_p}{Q_a} \quad (2.17)$$

Para determinar los parámetros D_1 y D_2 . Posteriormente, se emplea la expresión (2.2) para determinar la cobertura solar para cada mes, f . De este modo, la energía

útil captada cada mes por el colector, Q_u , viene dada por la expresión:

$$Q_u = f \cdot Q_a \quad (2.18)$$

Y la cobertura solar anual viene dada por la expresión:

$$\text{Cobertura solar anual} = \frac{\sum_{u=1}^{12} Q_u}{\sum_{a=1}^{12} Q_a} \quad (2.19)$$

2.2. Programación e implementación en MATLAB del método de cálculo seleccionado

En esta sección se va a explicar el procedimiento seguido para la programación del método de cálculo. Tan sólo se incluye en este apartado las líneas de código más relevantes y la estructura seguida en la programación, dejándose para el anexo correspondiente todo el código.

Es necesario comenzar por explicar la variable de tipo estructura (struct) denominada `municipios`. Partiendo de la base de datos procedente de [31], se ha generado la variable `municipios`, con los siguientes campos:

- `ind` (integer): índice del municipio según la base de datos [31]
- `provincia` (string): Provincia a la que pertenece el municipio
- `municipio` (string): Nombre del municipio
- `ref` (string): nombre del archivo de texto (.txt) generado por PVGIS con los datos de radiación solar
- `altura` (double): Altura del municipio en metros sobre el nivel del mar
- `capital` (string): Capital de provincia del municipio. Coincidirá con la provincia en caso de que tenga el mismo nombre la capital de provincia que la provincia (i.e. Jaén) y no coincidirá en otros casos (Vigo - Pontevedra).
- `PVGIS` (struct): estructura con los datos procedentes del archivo de texto de PVGIS, generado a partir de estos con la función `LeerDatosPVGIS.m` (Anexo A).

- lat (double): Latitud en formato decimal
- long (double): Longitud en formato decimal

El primer paso es solicitar la provincia en la que se encuentra el proyecto. A partir de esta variable, o más bien base de datos, una vez seleccionada la provincia, se busca en toda la base de datos los municipios que pertenecen a esa provincia, y los muestra en el siguiente menú desplegable, para que el usuario seleccione el municipio.

El segundo paso es la determinación de la demanda de ACS a 60°C, según la Tabla 2.2. Para ello, la tabla anteriormente citada se ha guardado en tres variables, a saber: Edificios, Gy Consumos, siendo cada variable la columna primera, segunda y tercera respectivamente de la tabla. Así, la *Graphical User Interface* (GUI), solicita al usuario el número de unidades de consumo (personas, camas, etc...) para el edificio que el usuario ha seleccionado (Hospital, residencia, vivienda, etc...), y los multiplica por el número de litros correspondientes.

En caso de que se trate de una vivienda (unifamiliar o multifamiliar), será necesario realizar la conversión de dormitorios a personas, ya que la GUI solicita número de dormitorios. Para ello, se emplea la función `dorm2pax`:

```
1      function personas = dorm2pax(dormitorios)
2      %% funcion : Descripcion
3      % Dado un numero de dormitorios, devuelve el numero de ...
4      %      personas estimado
5      % segun el CTE
6      %
7      % Parametros de entrada :
8      % - dormitorios: numero de dormitorios
9      %
10     % Valores de retorno :
11     % - personas: numero de personas
12     %
13     % Precondiciones
14     % - ninguna
15     % Trabajo Fin de Grado. Ingenieria de Recursos Energeticos
```

```

16      % Manuel G. Alcazar Vargas. 2018
17
18      x = 1:1000;
19      x(1:7) = [1.5 3 4 6 7 8 9];
20      personas = x(dormitorios);
21      end

```

Para poder determinar la demanda corregida, es necesario emplear la ecuación (2.7). La altitud de los municipios se ha obtenido de [31]. Así, se emplea la función Tred2Tred:

```

1      function municipioTred = ...
           Tred2Tred(capitalTred,capitalAltura,municipioAltura)
2      % Funcion que dada la temperatura de red de la capital de ...
           provincia [T_AFCP] (Tabla
3      % B.1 CTE) y la altura, tanto de la capital de provincia como ...
           del municipio
4      % en cuestion, devuelve la temperatura media mensual de agua ...
           fria del municipio en cuestion [T_AFY]
5      % Az = Altura de la localidad - Altura de la capital de provincia
6      % T_AFY = T_AFCP - B*Az
7      % Donde
8      %      B = 0,010 para los meses de octubre a marzo;
9      %      B = 0,005 para los meses de abril a septiembre.
10
11      Az = municipioAltura - capitalAltura;
12      B = 0.010*ones(1,12);
13      B(4:9) = 0.005;
14      municipioTred = capitalTred - B.*Az;
15      end

```

Y se calcula la demanda corregida:

```

1      % Se calcula la temperatura media del agua de la red segun el CTE
2      datosMunicipio.Tred = ...
           Tred2Tred(datosCapital.Tred,datosCapital.altura,datosMunicipio.altura);
3

```

```
4      % Calculo de la demanda
5      contribucion = dem2contrib(demanda,datosMunicipio.PVGIS.zona);
6      Tacum = str2double(get(handles.editTacum,'String')); % degC
7      Tacs = str2double(get(handles.editTacs,'String')); % degC
8      rho = str2double(get(handles.editRho,'String')); % kg/L
9      cp = str2double(get(handles.editCp,'String')); % kJ/kg/degC
10     tamano = get(handles.popupmenuTamano,'Value');
11     material = get(handles.popupmenuMaterial,'Value');
12     ndias = [31 28 31 30 31 30 31 31 30 31 30 31]';
13     Di = demanda.*((60 - datosMunicipio.Tred)./(Tacs - ...
        datosMunicipio.Tred));
```

Con la demanda corregida y la zona, se entra en la tabla 2.5 (el programa no está hecho para el caso en que la fuente de energía auxiliar sea eléctrica) y se determina la contribución solar mínima.

El resto de los cálculos son idénticos a los descritos en la sección anterior, y se pueden consultar en el anexo B.

2.3. Elaboración de una herramienta de cálculo virtual a través de una GUI en MATLAB intuitiva para el cálculo de instalaciones solares térmicas de BT

En esta sección se explica el funcionamiento de la *Graphical User Interface* (GUI) elaborada en el presente TFG y que es el núcleo del mismo. Se ha programado con el fin de que pueda ser una herramienta didáctica y, por tanto, intuitiva.

El código se encuentra en el anexo correspondiente (Anexo B), mientras que la explicación del funcionamiento de la misma se realiza en el próximo capítulo. Aquí se desarrolla la estructura y los tipos de objetos que están presentes en la GUI, fundamentalmente.

A la GUI se puede acceder desde el entorno de MATLAB «guide» que nos permite modificarla (Figura 2.3), o bien desde el entorno de usuario, que nos permite utilizarla (Figura 2.4). Para entrar en el entorno de «guide» será necesario escribir

en la ventana de comandos de MATLAB el comando `guide`, y posteriormente nos ofrece o crear una GUI desde cero, emplear algunas de las plantillas que trae por defecto MATLAB o abrir una GUI existente (Figura 2.5).

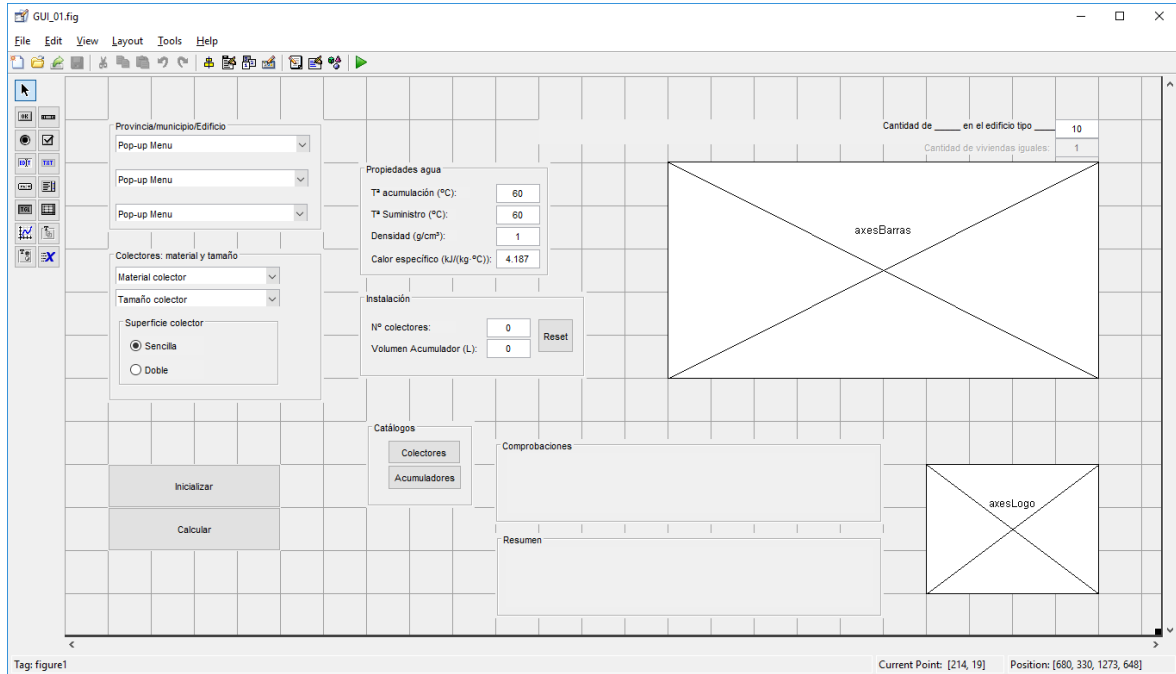


Figura 2.3: Vista de la GUI desde el entorno de MATLAB «guide»

Cada elemento de la GUI tiene asociado un nombre (*tag*) y una función. Algunos realizan no nada, como por ejemplo, los *static text* que simplemente indican que a lo que hace referencia el cuadro adyacente. Otros, como los botones (*pushbutton*) sí ejecutan líneas de código. Cuando se pulsa un botón, se ejecutan las líneas de código asociadas al elemento. En el caso del botón *Reset*, pone a cero el número de colectores y el volumen de acumulación (líneas 6 y 7):

```

1  % --- Executes on button press in pushbuttonReset.
2  function pushbuttonReset_Callback(hObject, eventdata, handles)
3  % hObject    handle to pushbuttonReset (see GCBO)
4  % eventdata  reserved - to be defined in a future version of MATLAB
5  % handles    structure with handles and user data (see GUIDATA)
6  set(handles.editV, 'String', '0');
7  set(handles.editn, 'String', '0');

```

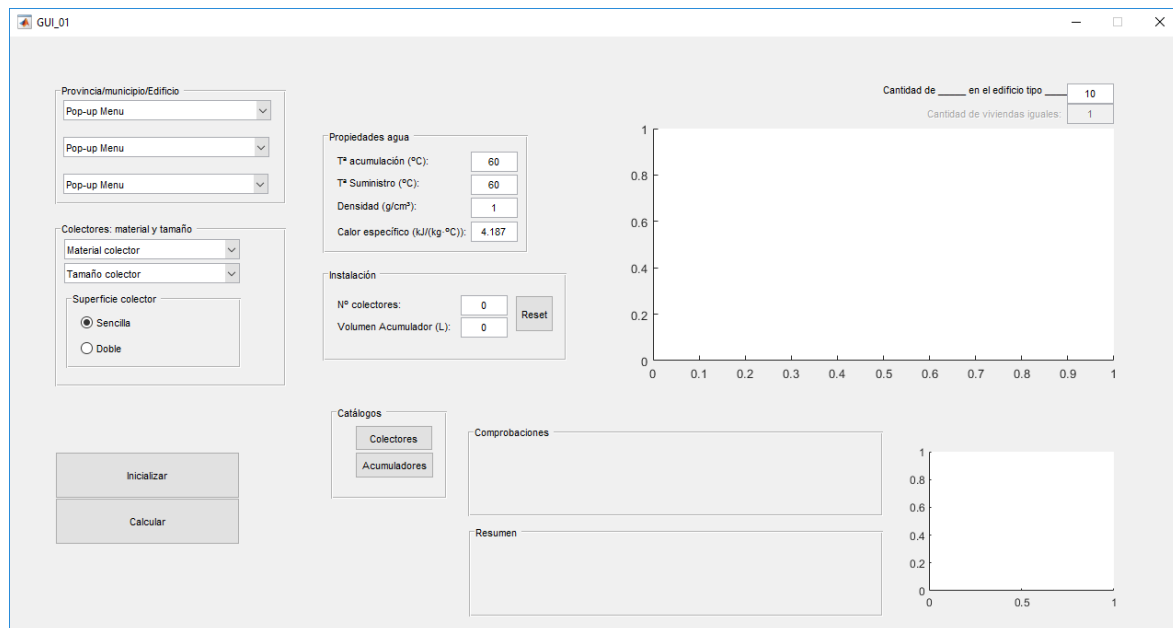
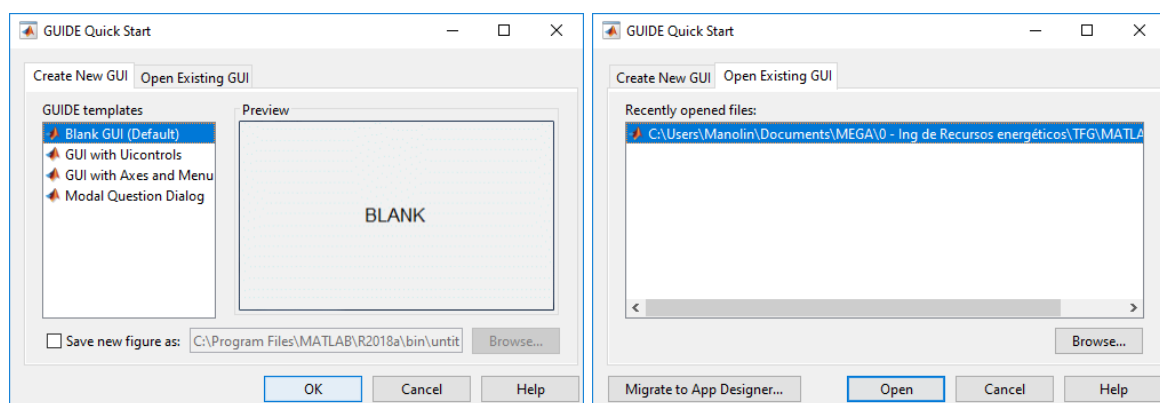


Figura 2.4: Vista de la GUI desde el entorno de MATLAB de usuario



(a) Crear una GUI nueva

(b) Abrir una GUI existente

Figura 2.5: Opciones para abrir una GUI desde el entorno *guide*

Capítulo 3

Análisis de los resultados

En este capítulo, se va emplear la aplicación desarrollada para el cálculo de diversos edificios. En concreto, se calcularán los mismos que se describen en [12], para poder posteriormente validar el método. Los edificios que se van a estudiar son los siguientes:

- Hotel **** en Miraflores de la Sierra (Madrid) con capacidad de 100 personas
- Vivienda unifamiliar en Sevilla de 8 personas
- Centro polideportivo en Jaén con capacidad para 350 personas
- Residencia geriátrica en Tudela (Navarra) con capacidad para 198 personas

3.1. Aplicación de la herramienta de cálculo en diferentes escenarios

En esta sección, se explica el funcionamiento de la herramienta desarrollada. Para ello, se abre MATLAB y se ejecuta la GUI (`CalculadoraSolarGUI.m`). Aparecerá la pantalla que se muestra en la Figura 3.1.

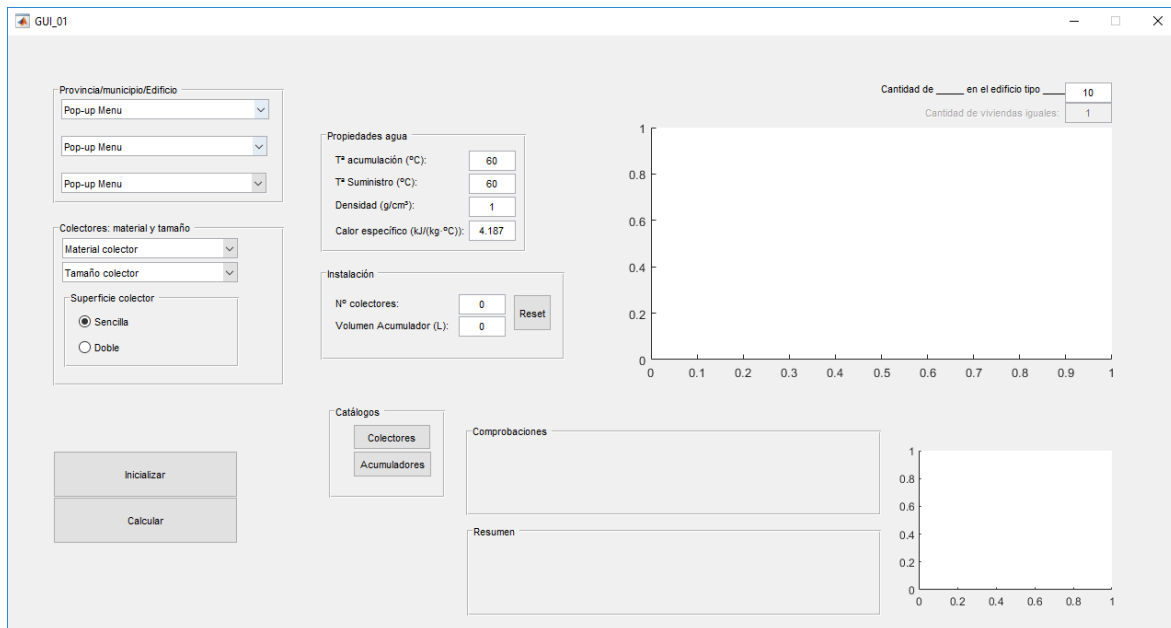


Figura 3.1: Ventana de inicio de la GUI

Lo siguiente será hacer clic en «Iniciar» para que se carguen las variables con las que trabaja la aplicación (Figura 3.2):

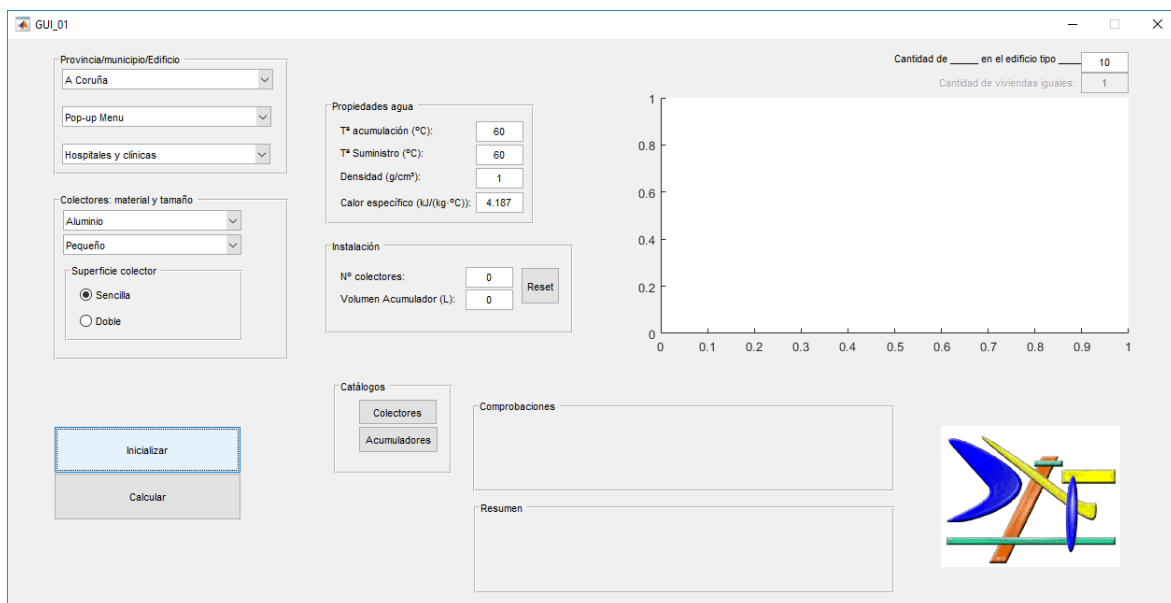


Figura 3.2: Ventana de trabajo de la GUI

El siguiente paso, para calcular el hotel, será seleccionar la provincia de Madrid, el municipio de Miraflores de la Sierra (Figura 3.3) y en el tipo de edificio, hotel ****. Una vez se selecciona el tipo de edificio, el programa solicita el número de unidades, según la tabla 2.2 (Figura 3.4)

Figura 3.3: Paso 1: Provincia y municipio

Figura 3.4: Paso 2: Tipo de edificio y número de unidades de consumo

En este caso, el IDAE habla de un hotel de 100 personas y el CTE considera el número de camas del hotel. Supondremos que 80 camas equivalen a 100 personas. Puesto que es una instalación grande, se emplearán colectores solares grandes.

Las propiedades del agua no se modifican, pues no se indica nada de que la temperatura de almacenamiento o de distribución del ACS sea diferente de 60°C que es lo que se considera estándar. Tampoco se modifica ni la densidad ni el calor específico del agua. En cuanto a la superficie de los colectores, se supone que es

sencilla.

El siguiente paso será hacer clic en «Calcular» para que nos indique el número de colectores y el volumen de acumulación (Figura 3.5).

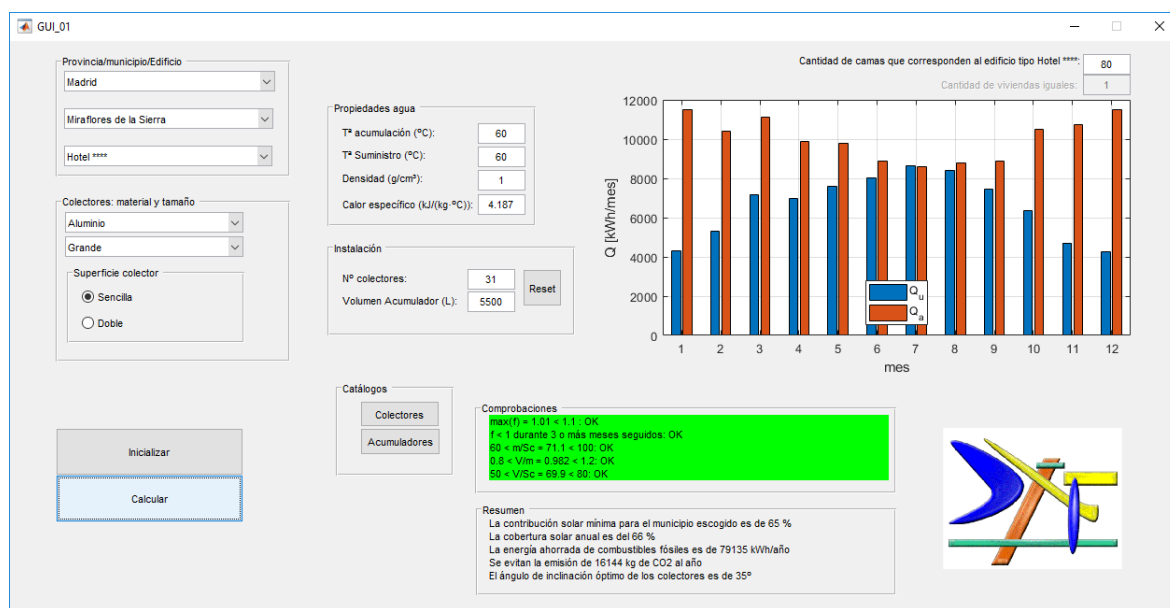


Figura 3.5: Paso 3: Calcular

El programa arroja un resultado de 31 colectores solares y un volumen de acumulación de 5.500 L. Una configuración con un número impar de colectores no es recomendable, y menos aún si es un número primo. Modificaremos el número a 32 (que es 2^5) y el volumen de acumulación lo reduciremos a 5.000 L ya que la demanda en un hotel es más o menos continua a lo largo del día y no es necesario almacenar un volumen tan elevado. Además, en el catálogo de acumuladores que se está considerando en este TFG se encuentra un colector de 5.000 L. Una vez modificados los datos, volvemos a calcular (Figura 3.6):

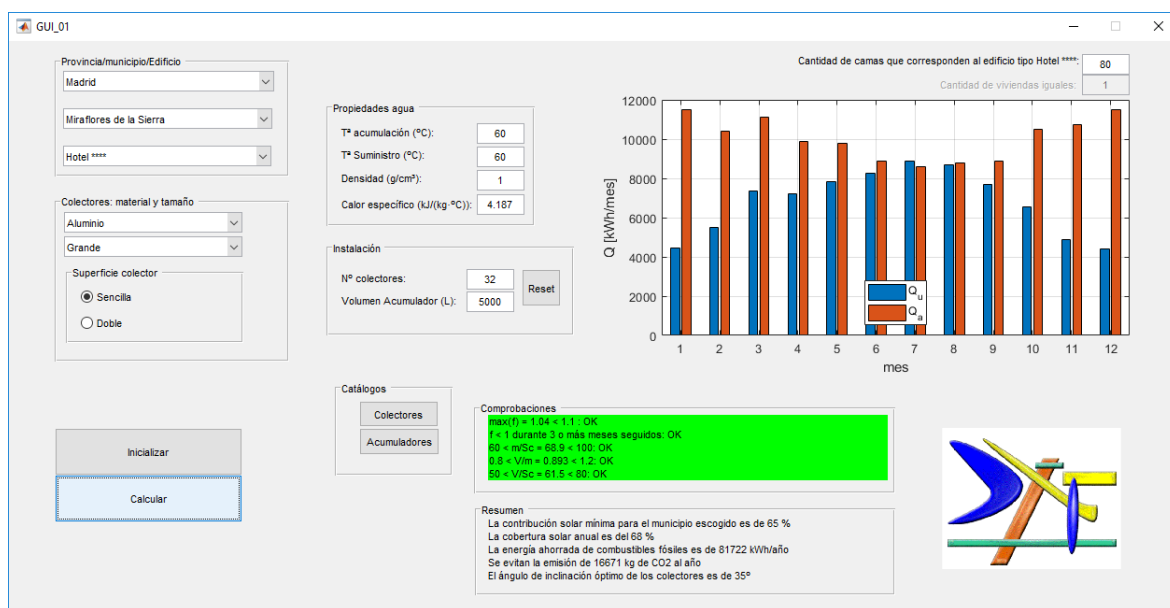


Figura 3.6: Paso 4: Recalcular

Se observa que en este caso también se cumplen todos los condicionantes, empleando 32 colectores solares con una superficie de captación unitaria de 2,46 m² se tiene una superficie total de captación de 78,72 m² y una fracción solar del 68%; frente a los 80 m² y una fracción solar del 74 %. En caso de emplear 34 colectores solares, se obtendría una superficie total de captación de 83,64 m² y una fracción solar del 72 %.

En el segundo caso, se va a calcular una vivienda unifamiliar de 8 personas con una demanda de 40 L/persona·día. La Tabla 2.2 considera en el caso de viviendas unifamiliares una demanda de 30 L/persona·día; de modo que esta vivienda unifamiliar sería equivalente a una vivienda de unas 11 personas, en Sevilla (Figura 3.7):.

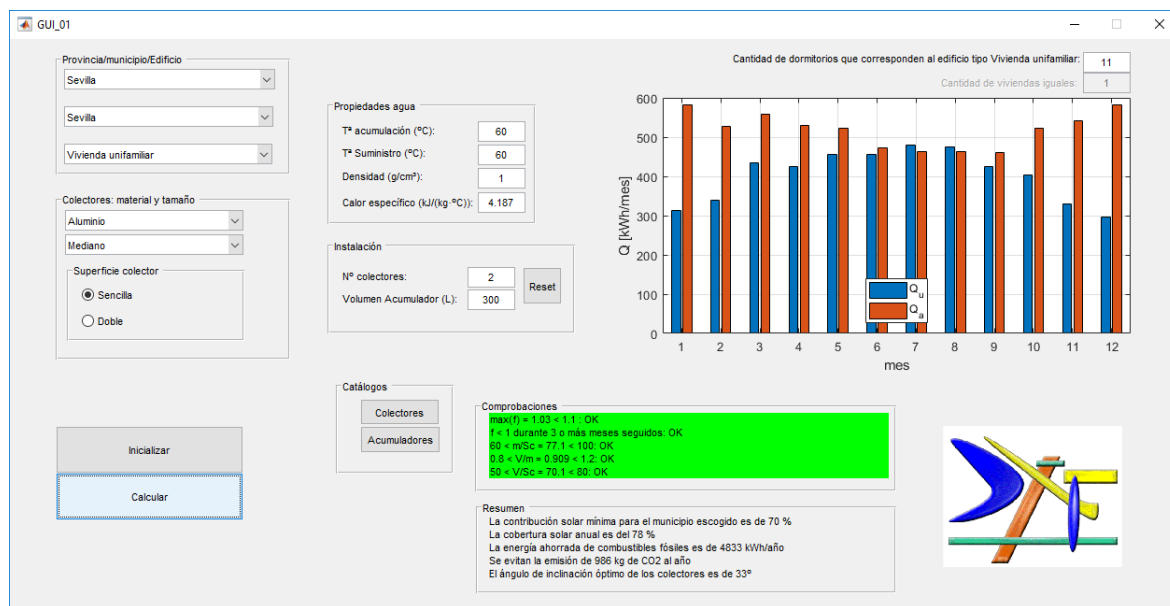


Figura 3.7: Caso 2: Vivienda unifamiliar en Sevilla

En este caso, con colectores pequeños no se podía realizar puesto haría falta un número intermedio entre dos y tres que no existe. En el caso de colectores solares grandes ocurre lo mismo entre uno y dos, de modo que es necesario emplear dos colectores de tamaño mediano, con una superficie de captación unitaria de $2,14 \text{ m}^2$.

El tercer caso es un polideportivo en Jaén. El CTE considera una demanda de 25 L/usuario·día, mientras que en el proyecto se consideraron 350 personas con una demanda de 20 L/usuario·día. Por tanto, en la aplicación deberemos indicar que equivale a 280 personas. Al ser una instalación grande, se emplearán colectores solares grandes. Al calcular, el programa arroja un número de colectores de 39 (Figura 3.8), pero al ser la radiación solar en verano en Jaén muy alta, se obtiene una f máxima superior a 1,10; así como durante los meses de verano (junio, julio y agosto) se supera $f > 1$. Por tanto, será necesario reducir el número de colectores solares. Si se disponen 36 colectores y un volumen de acumulación de 6.000 L (dos depósitos de 3.000 L) ya cumple los requisitos (Figura 3.9).

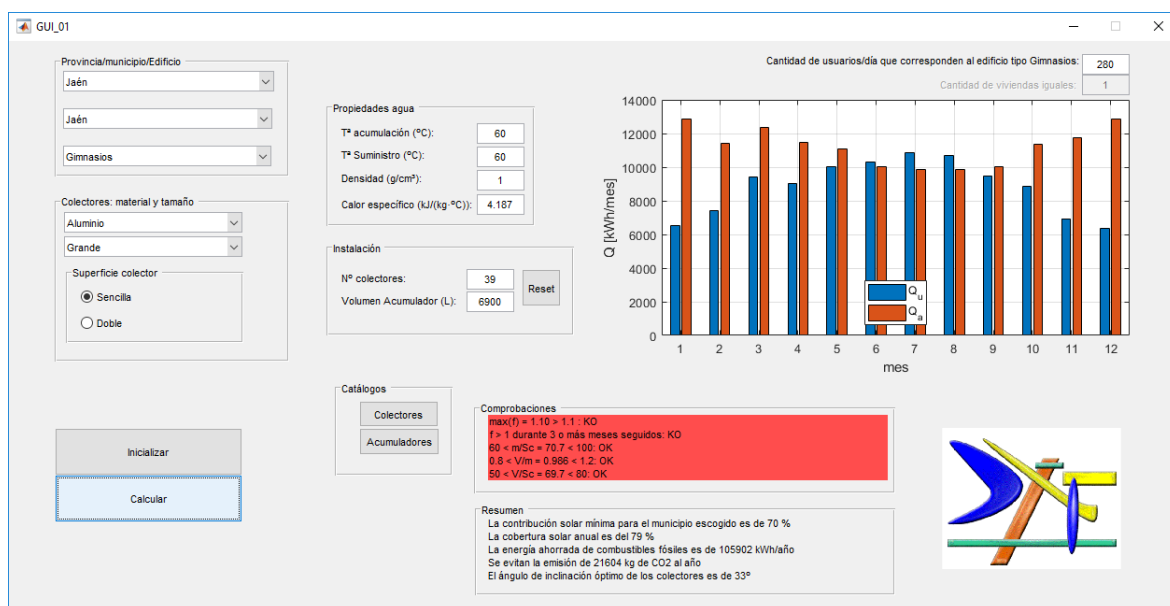


Figura 3.8: Caso 3: Polideportivo en Jaén. Excesiva superficie de captación

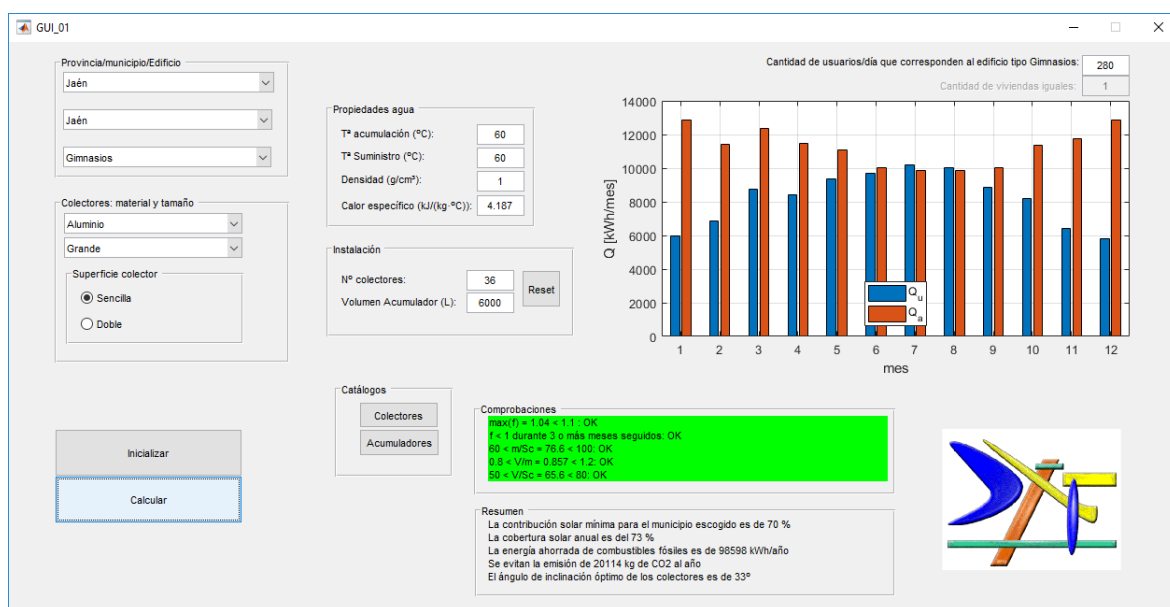


Figura 3.9: Caso 3: Polideportivo en Jaén. Superficie de captación adecuada

El último caso que aquí se considera es una residencia de ancianos en Tudela (Navarra). En este caso la demanda unitaria no coincide con la que indica el CTE, siendo la proyectada 66 L/persona·día mientras que el CTE propone 55 L/persona·día. De este modo, las 198 personas que se estiman en la residencia serían equivalente a 238 personas. En este caso, el programa propone emplear 74 colectores solares grandes y un volumen de acumulación de 13.200 L, que nosotros modificaremos a 15.000 L para disponer tres depósitos de 5.000 L, puesto que en

Navarra sí es probable tener días sin sol y se puede esperar diferencia temporal entre el consumo y la generación del agua caliente sanitaria.

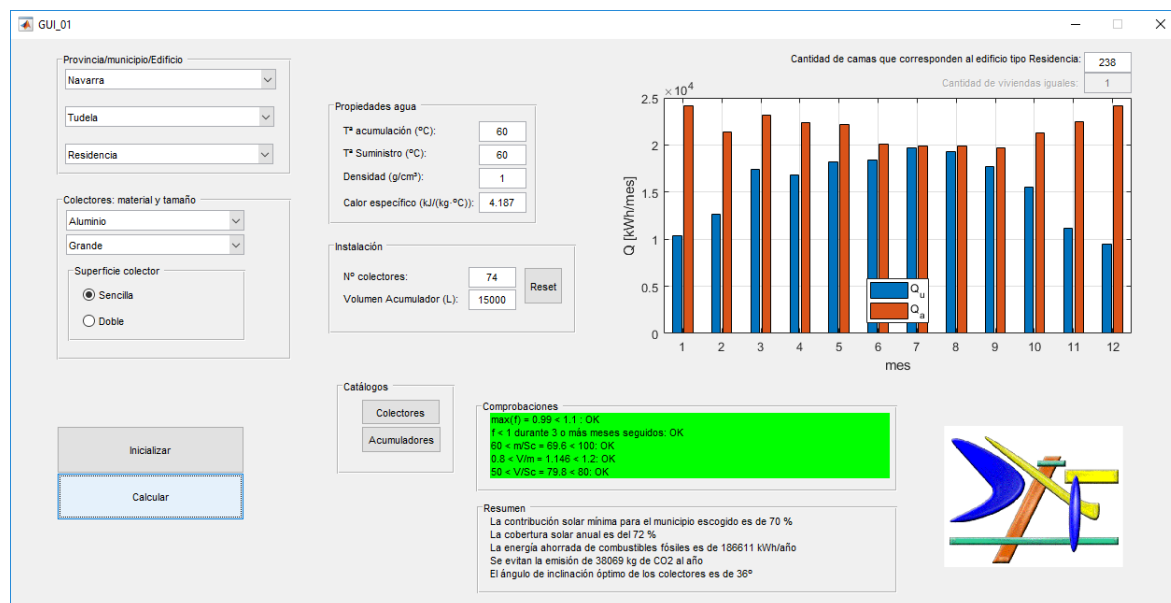


Figura 3.10: Caso 4: Residencia de ancianos en Tudela (Navarra)

3.2. Validación y análisis de los resultados

En esta sección se va a comprobar el funcionamiento de la aplicación desarrollada. Para ello, se comparan los resultados obtenidos por ésta con las instalaciones solares ya existentes descritas anteriormente y que se pueden encontrar en [12]. La Tabla 3.1 muestra los resultados.

Tabla 3.1: Comparativa entre los resultados procedentes de la aplicación e instalaciones solares ya existentes. Fuente elaboración propia y [12]

	IDAE		GUI	
	S_c (m²)	f (%)	S_c (m²)	f (%)
Hotel ****	80,00	74	83,64	72
Vivienda unifamiliar	4,34	80	4,28	78
Polideportivo	81,00	72	88,56	73
Residencia ancianos	247,80	65	182,04	72

Se observa que para los tres primeros casos los resultados que se obtienen son prácticamente idénticos, mientras que en el último hay una discrepancia considerable. Si se busca información sobre el edificio, la única residencia de ancianos de Tudela que reúne esas características es la *Real Casa de la Misericordia*

(<http://www.rcmisericordia.com/>). En una ortofotografía actual se puede ver la distribución de los colectores solares (Figura 3.11)

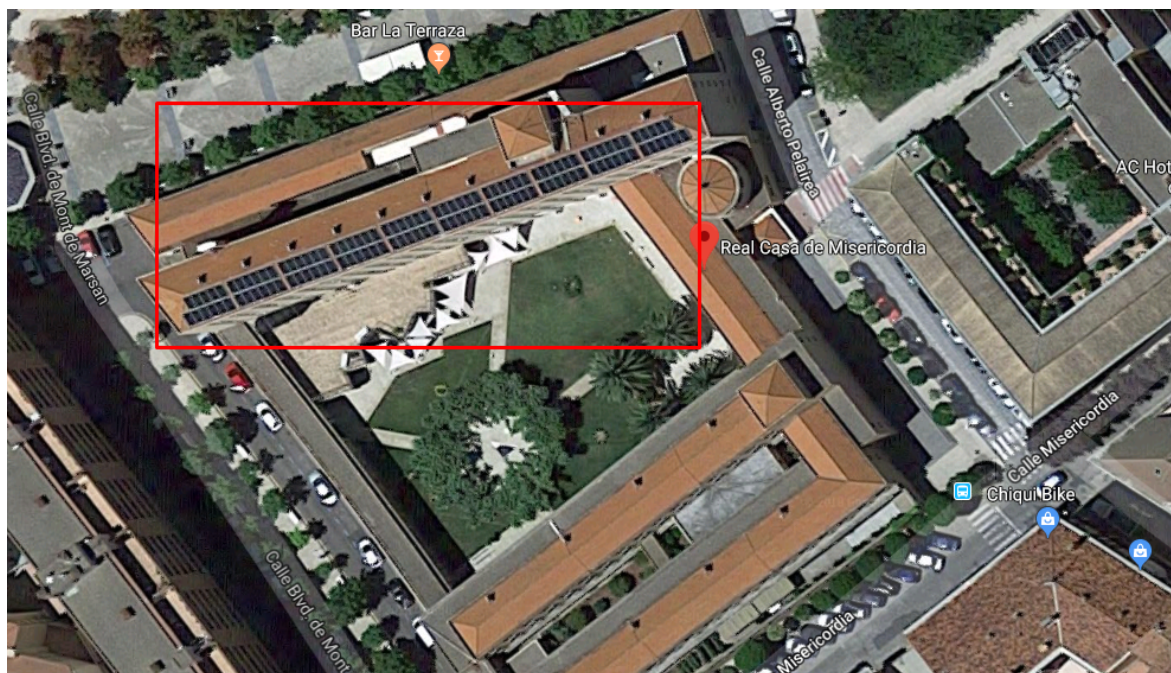


Figura 3.11: Ortofotografía de la Real Casa de la Misericordia en Tudela (Navarra). Fuente: Google Maps

Se observa que los colectores no presentan orientación totalmente al sur sino más bien al sureste, además de que la inclinación de los mismos no es la óptima (36°), sino la de la cubierta, que es desconocida por el autor de este trabajo. Además, la cobertura solar según [12] es del 65 % mientras que el CTE indica que para este tipo de edificaciones, con esa demanda y en esa zona debería ser de un 70 %, lo que sugiere que la instalación es antigua y, por tanto, se realizó con tecnología de menor rendimiento.

Todos estos factores motivan que con una superficie mayor, se consiga menor energía térmica, aunque realmente se desconoce el por qué de la discrepancia en los resultados.

En el resto de instalaciones, como ya se ha comentado, los resultados coinciden, **validando, por tanto, la aplicación desarrollada en el presente TFG**

Capítulo 4

Conclusiones finales y propuesta de mejora

El trabajo fin de grado desarrollado comienza con una introducción histórica a la energía solar, haciendo hincapié en las aplicaciones que tiene esta energía en aplicaciones con fines térmicos, para continuar describiendo brevemente los tipos y partes de instalaciones solares térmicas de baja temperatura.

En ese mismo capítulo se ha realizado una breve revisión de los métodos de cálculo actualmente empleados, siendo el más relevante el *f-chart*. Posteriormente, en el capítulo 2 se desarrolla matemáticamente el método empleado y se explica, someramente, cómo se ha programado en el software utilizado, que es MATLAB. Se deja para el Anexo B el código completo de la GUI. A continuación, se ha explicado cuál es la estructura de la GUI, tanto desde el punto de vista visual, como de las variables y bases de datos que emplea.

En el tercer capítulo se realiza, por un lado un *tutorial* en el que se explica cómo utilizar la aplicación, así como una serie de ejemplos, que se consideran suficientemente variados pues incluyen desde las aplicaciones de menor envergadura posibles (vivienda unifamiliar) hasta grandes instalaciones, además de estar geográficamente situadas a lo largo y ancho de la geografía española. También se han utilizado diferentes tipos de colectores solares, según el tamaño de la instalación.

Continuando en el mismo capítulo, se contrastan los resultados obtenidos por la aplicación con los que propone el Instituto para la Diversificación y el Ahorro de Energía (IDAE) en su manual de energía solar térmica [12]. Se puede comprobar

que los resultados que ofrece la aplicación son más que aceptables, al no observar prácticamente discrepancia entre las instalaciones reales y las aquí propuestas.

A continuación, se mencionan las posibles mejoras que podrían incluirse en la aplicación y que no se han incluido bien porque se podrían considerar fuera del alcance del trabajo fin de grado o bien por motivos de tiempo.

- Quizás el punto más interesante de cara a la ampliación del TFG sería la inclusión del **resto de la instalación**, a saber:
 - Cálculo del intercambiador de calor entre el fluido de trabajo y el ACS
 - Determinación del caudal del fluido de trabajo, diámetro de tuberías y pérdidas de carga
 - Cálculo de la bomba hidráulica
 - Determinación del espesor del aislamiento de las tuberías
 - Selección del vaso de expansión
 - Disposición de los colectores: número en serie y paralelo, separación mínima, etc...
- Se podría incluir también un **estudio económico** para poder seleccionar la alternativa más económica
- El programa calcula siempre con la inclinación óptima según el PVGIS y al seleccionar un municipio se toma la posición procedente de la base de datos de [31]. En ciudades como Madrid puede haber una diferencia considerable entre un punto y otro. Podría incluirse una opción de seleccionar un archivo de texto procedente de PVGIS que introduzca el usuario, con la **inclinación de los colectores** que se considere en el proyecto y en la **ubicación exacta**.
- No se ha considerado el caso de climatización en **piscinas**
- Tampoco se ha considerado la alternativa de que la **fuentes auxiliar de calor sea eléctrica**, modificándose, por tanto, la contribución mínima necesaria.
- Al generar la base de datos de los municipios, en el entorno de PVGIS Europa no se encuentran las **Islas Canarias**, de modo que no fueron incluidas.

Posteriormente el tutor del TFG indicó que los datos para las Islas Canarias se encuentran en el entorno de PVGIS África. No se han incluido por dos motivos: puesto que la aplicación tiene fines didácticos no hay diferencia y por falta de tiempo.

- Se pueden ampliar los **catálogos**, tanto de volumen de acumulación como de colectores

Anexos

Apéndice A

PVGIS

En este apartado se explica el funcionamiento de la página web *PVGIS* (<http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php?lang=es&map=europe>). *PVGIS* es una base de datos que permite obtener la radiación solar incidente sobre un punto, con fines de energía solar fotovoltaica o térmica, entre otros. Ésta es una herramienta potentísima pues evita el tedioso cálculo de la radiación solar sobre un punto y los datos que proporciona son muchísimo más precisos que los estimados por otros métodos de cálculo. Para más información sobre el proyecto *PVGIS*, se puede consultar [32].

Para el caso que nos ocupa en el presente TFG: energía solar térmica de baja temperatura, será necesario seleccionar la pestaña correspondiente a *radiación mensual* (Figura A.1).

JRC CM SAF Sistema de Información geográfica fotovoltaica - mapa interactivo

EUROPA > CE > CCI > IET > RE > SOLAREC > PVGIS > Mapa interactivo > Europa

Por ejemplo, "Ispra, Italy" "45.256N, 16.9589E"

posición del cursor: 45.150, 5.475
posición elegida: 47.559, 18.064

Latitud: Longitud:

Francia

Datos irradiación global mensual

Base de datos de radiación: Climate-SAF PVGIS

☒ Irradiación horizontal
☒ Irradiación con el ángulo óptimo
☐ Irradiación directa normal
☒ Irradiación sobre el ángulo seleccionado: 90 grados
☐ Turbidez de Linke
☐ Radiación dif./global
☒ Ángulo de inclinación óptimo

Datos de temperatura ambiente mensual

☐ Temperatura media del día
☒ Media diaria de temperatura
☒ Número de grados día de calefacción

Formatos de salida

☐ Mostrar gráficas ☐ Mostrar el horizonte
☐ Página web ☐ Fichero de texto ☐ PDF

[\[ayuda\]](#)

Figura A.1: Radiación mensual PVGIS

Posteriormente, habrá que seleccionar el municipio, en este caso Linares, y marcar todas las casillas. Además, queremos que nos devuelva los datos procedentes de la base de datos en texto plano (.txt) para poder leerlos posteriormente con MATLAB. La página web debería quedar como la Figura A.2.

JRC CM SAF Sistema de Información geográfica fotovoltaica - mapa interactivo

EUROPA > CE > CCI > IET > RE > SOLAREC > PVGIS > Mapa interactivo > Europa

Por ejemplo, "Ispra, Italy" "45.256N, 16.9589E"

posición del cursor: 38.099, -3.485
posición elegida: 38.097, -3.637

Latitud: Longitud:

Linares

Datos irradiación global mensual

Base de datos de radiación: Climate-SAF PVGIS

☒ Irradiación horizontal
☒ Irradiación con el ángulo óptimo
☒ Irradiación directa normal
☒ Irradiación sobre el ángulo seleccionado: 90 grados
☒ Turbidez de Linke
☐ Radiación dif./global
☒ Ángulo de inclinación óptimo

Datos de temperatura ambiente mensual

☒ Temperatura media del día
☒ Media diaria de temperatura
☒ Número de grados día de calefacción

Formatos de salida

☐ Mostrar gráficas ☐ Mostrar el horizonte
☐ Página web ☒ Fichero de texto ☐ PDF

[\[ayuda\]](#)

Figura A.2: Configuración para descargar la radiación solar

A continuación, se hace clic en calcular, y se obtiene un archivo de texto como

el de la Figura A.3.

MonthlyRadiation380548N_033813W.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

Latitud: 38°5'48" Norte,
Longitud: 3°38'13" Oeste

El ángulo de inclinación óptimo es: 34 grados

Irradiación anual perdida a causa de las sombras (horizontal): 0.0 %

Mes	Hh	Hopt	H(90)	DNI	Iopt	TL	D/G	TD	T24h	NDD
Ene	2370	3960	3970	3700	62	2.6	0.41	10.0	7.8	285
Feb	3400	5030	4460	4760	55	2.9	0.35	10.3	8.2	206
Mar	4800	5950	4270	5320	41	2.7	0.39	14.1	11.8	116
Abr	5620	6010	3250	5710	25	3.4	0.35	18.0	15.9	71
Mayo	6790	6500	2650	6740	13	3.7	0.32	22.2	20.1	3
Jun	7940	7180	2320	8590	4	4.1	0.24	26.6	24.7	0
Jul	8150	7550	2540	9350	8	4.1	0.20	30.8	28.7	0
Ago	7140	7350	3380	8250	21	4.3	0.22	30.7	28.6	0
Sep	5420	6460	4170	6490	36	4.0	0.28	25.7	23.5	9
Oct	4070	5660	4670	5310	50	3.0	0.35	21.1	18.5	71
Nov	2700	4390	4260	4180	60	2.8	0.37	14.5	12.2	258
Dic	2190	3860	4030	3680	65	2.5	0.41	11.2	8.8	285
Año	5860	5830	3660	6010	34	3.3	0.30	19.6	17.4	1304

Hh: Irradiación sobre plano horizontal (Wh/m2/día)
Hopt: Irradiación sobre un plano con la inclinación óptima (Wh/m2/día)
H(90): Irradiación sobre plano inclinado:90grados (Wh/m2/día)
DNI: Irradiación directa normal (Wh/m2/día)
Iopt: Inclinación óptima (grados)
TL: Turbidez de Linke (-)
D/G: Ratio entre la irradiación difusa y la global (-)
TD: Temperatura media del día (°C)
T24h: Temperatura media diaria (24h) (°C)
NDD: Número de grados día de calefacción (-)

PVGIS (c) Comunidades europeas, 2001-2012

Figura A.3: Archivo de texto generado por la página PVGIS

Lo siguiente, será leer esos datos con MATLAB, empleando la función que se muestra a continuación y generar una estructura con los campos siguientes:

- Latitud (double)
- AnguloOptimo (double)
- PerdidasSombras (double)
- Hh (12 × 1 double)
- Hopt (12 × 1 double)
- H90 (12 × 1 double)
- lopt (12 × 1 double)
- TL (12 × 1 double)
- DG (12 × 1 double)
- Td (12 × 1 double)
- T24h (12 × 1 double)
- NDD (12 × 1 double)
- mes (12 × 1 cell)
- zona (integer)

```

1 function Output = LeerDatosPVGIS(filename)
2
3 %% funcion : Descripcion
4 % Dado un archivo de texto plano, devuelve una estructura con los ...
   datos
5 %
6 % Parametros de entrada :
```

```
7  % - filename: nombre del fichero de texto plano. Si esta vacio, se ...
    busca
8  %
9  % Valores de retorno :
10 % - PVGGIS_data: Datos PVGIS interpretados por MATLAB
11 %
12 % Precondiciones
13 % - ninguna
14 %
15 % Trabajo Fin de Grado. Ingenieria de Recursos Energeticos
16 % Manuel G. Alcazar Vargas. 2018
17
18 if nargin == 0 % Si no se pasa ningun parametro de entrada
19     [filename,pathname] = uigetfile('*.txt');
20     filename = [pathname filename];
21 end
22     fid = fopen(filename,'r');
23     i = [1,3,4,7:19]; % Lineas que quiero leer
24     tline = fgetl(fid);
25     j = 1;
26     k = 1;
27     A = zeros(13,10);
28 while ischar(tline)
29     if sum(j==i) == 1 % Si esa linea la quiero leer
30         if j == 1 % Si es la latitud
31             temp = tline;
32             temp = strsplit(temp);
33             temp = temp{2}; % Latitud
34             lat = leerlatitud(temp);
35         elseif j == 3 % Si es el angulo optimo
36             temp = tline;
37             temp = strsplit(temp);
38             op = str2double(temp{7}); % angulo optimo
39         elseif j == 4 % Si es las perdidas por sombras
40             temp = tline;
41             temp = strsplit(temp);
42             som = str2double(temp{10}); % Perdidas por sombras
43         else
44             temp = tline;
```

```

45         temp = strsplit(temp);
46         for l = 2:11
47             A(k,l-1) = str2double(temp{l});
48         end
49         k = k+1;
50     end
51 end
52 j = j+1;
53 tline = fgetl(fid);
54 end
55
56 fclose(fid);
57 % Output.Nombre = nombre;
58 Output.Latitud = lat;
59 Output.AnguloOptimo = op;
60 Output.PerdidasSombras = som/100;
61 Output.Hh = A(1:12,1);
62 Output.Hopt = A(1:12,2);
63 Output.H90 = A(1:12,3);
64 Output.DNI = A(1:12,4);
65 Output.Iopt = A(1:12,5);
66 Output.TL = A(1:12,6);
67 Output.DG = A(1:12,7);
68 Output.Td = A(1:12,8);
69 Output.T24h = A(1:12,9);
70 Output.NDD = A(1:12,10);
71 Output.mes = {'Enero' 'Febrero' 'Marzo' 'Abril' 'Mayo' ...
               'Junio' 'Julio' 'Agosto' 'Septiembre' 'Octubre' 'Noviembre' ...
               'Diciembre'};
72 Output.zona = kWh2zona(A(13,1));
73 end
74
75 function lat = leerlatitud(s)
76 % funcion auxiliar que lee la latitud
77 s = double(s);
78 p1 = find(s == double(176)); % Posicion elemento 'deg'
79 p2 = find(s == double( 39)); % Posicion elemento ''
80 p3 = find(s == double( 34)); % Posicion elemento '"'
81 deg = s( 1:p1-1);

```

```
82     min = s(p1+1:p2-1);
83     sec = s(p2+1:p3-1);
84     deg = str2double(char(deg));
85     min = str2double(char(min));
86     sec = str2double(char(sec));
87     lat = dms2degrees([deg min sec]);    % Lo expresamos como ...
        [angulo minuto segundo]
88 end
89
90 function zona = kWh2zona(Hh)
91 % funcion auxiliar que determina la zona segun la tabla 3.2 del CTE
92     B = [0 3.8 4.2 4.6 5]*1000;
93     zona = 5;
94     for i = 1:length(B)-1
95         if (B(i) ≤ Hh && B(i+1) > Hh)
96             zona = i;
97         end
98     end
99 end
```

Apéndice B

Código GUI.m

```
1 function varargout = GUI_01(varargin)
2 % GUI_01 MATLAB code for GUI_01.fig
3 %     GUI_01, by itself, creates a new GUI_01 or raises the existing
4 %     singleton*.
5 %
6 %     H = GUI_01 returns the handle to a new GUI_01 or the handle to
7 %     the existing singleton*.
8 %
9 %     GUI_01('CALLBACK',hObject,eventData,handles,...) calls the ...
    local
10 %     function named CALLBACK in GUI_01.M with the given input ...
    arguments.
11 %
12 %     GUI_01('Property','Value',...) creates a new GUI_01 or ...
    raises the
13 %     existing singleton*. Starting from the left, property ...
    value pairs are
14 %     applied to the GUI before GUI_01_OpeningFcn gets called. An
15 %     unrecognized property name or invalid value makes property ...
    application
16 %     stop. All inputs are passed to GUI_01_OpeningFcn via varargin.
17 %
18 %     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows ...
    only one
19 %     instance to run (singleton)".
```

```

20 %
21 % See also: GUIDE, GUIDATA, GUIHANDLES
22
23 % Edit the above text to modify the response to help GUI_01
24
25 % Last Modified by GUIDE v2.5 21-Aug-2018 13:05:23
26
27 % Begin initialization code - DO NOT EDIT
28 gui_Singleton = 1;
29 gui_State = struct('gui_Name',       mfilename, ...
30                   'gui_Singleton',   gui_Singleton, ...
31                   'gui_OpeningFcn',   @GUI_01_OpeningFcn, ...
32                   'gui_OutputFcn',    @GUI_01_OutputFcn, ...
33                   'gui_LayoutFcn',    [] , ...
34                   'gui_Callback',     []);
35 if nargin && ischar(varargin{1})
36     gui_State.gui_Callback = str2func(varargin{1});
37 end
38
39 if narginout
40     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
41 else
42     gui_mainfcn(gui_State, varargin{:});
43 end
44 % load('inicializar.mat');
45 % End initialization code - DO NOT EDIT
46
47
48 % --- Executes just before GUI_01 is made visible.
49 function GUI_01_OpeningFcn(hObject, eventdata, handles, varargin)
50 % This function has no output args, see OutputFcn.
51 % hObject    handle to figure
52 % eventdata  reserved - to be defined in a future version of MATLAB
53 % handles    structure with handles and user data (see GUIDATA)
54 % varargin   command line arguments to GUI_01 (see VARARGIN)
55
56 % Choose default command line output for GUI_01
57 handles.output = hObject;
58

```

```
59 % Update handles structure
60 guidata(hObject, handles);
61
62 % UIWAIT makes GUI_01 wait for user response (see UIRESUME)
63 % uiwait(handles.figure1);
64
65
66 % --- Outputs from this function are returned to the command line.
67 function varargout = GUI_01_OutputFcn(hObject, eventdata, handles)
68 % varargout cell array for returning output args (see VARARGOUT);
69 % hObject handle to figure
70 % eventdata reserved - to be defined in a future version of MATLAB
71 % handles structure with handles and user data (see GUIDATA)
72
73 % Get default command line output from handles structure
74 varargout{1} = handles.output;
75
76
77 % --- Executes on selection change in listBoxProvincias.
78 function listBoxProvincias_Callback(hObject, eventdata, handles)
79 % hObject handle to listBoxProvincias (see GCBO)
80 % eventdata reserved - to be defined in a future version of MATLAB
81 % handles structure with handles and user data (see GUIDATA)
82
83 % Hints: contents = cellstr(get(hObject,'String')) returns ...
%         listBoxProvincias contents as cell array
84 %         contents{get(hObject,'Value')} returns selected item from ...
%         listBoxProvincias
85
86
87 % --- Executes during object creation, after setting all properties.
88 function listBoxProvincias_CreateFcn(hObject, eventdata, handles)
89 % hObject handle to listBoxProvincias (see GCBO)
90 % eventdata reserved - to be defined in a future version of MATLAB
91 % handles empty - handles not created until after all ...
%         CreateFcns called
92
93 % Hint: listbox controls usually have a white background on Windows.
94 %         See ISPC and COMPUTER.
```

```

95 if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
96     set(hObject,'BackgroundColor','white');
97 end
98
99
100 % --- Executes on selection change in listBoxMunicipio.
101 function listBoxMunicipio_Callback(hObject, eventdata, handles)
102 % hObject     handle to listBoxMunicipio (see GCBO)
103 % eventdata   reserved - to be defined in a future version of MATLAB
104 % handles     structure with handles and user data (see GUIDATA)
105
106 % Hints: contents = cellstr(get(hObject,'String')) returns ...
    listBoxMunicipio contents as cell array
107 %           contents{get(hObject,'Value')} returns selected item from ...
    listBoxMunicipio
108
109
110 % --- Executes during object creation, after setting all properties.
111 function listBoxMunicipio_CreateFcn(hObject, eventdata, handles)
112 % hObject     handle to listBoxMunicipio (see GCBO)
113 % eventdata   reserved - to be defined in a future version of MATLAB
114 % handles     empty - handles not created until after all ...
    CreateFcns called
115
116 % Hint: listbox controls usually have a white background on Windows.
117 %           See ISPC and COMPUTER.
118 if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
119     set(hObject,'BackgroundColor','white');
120 end
121
122
123 % --- Executes on selection change in listBoxEdificio.
124 function listBoxEdificio_Callback(hObject, eventdata, handles)
125 % hObject     handle to listBoxEdificio (see GCBO)
126 % eventdata   reserved - to be defined in a future version of MATLAB
127 % handles     structure with handles and user data (see GUIDATA)
128

```

```

129 % Hints: contents = cellstr(get(hObject,'String')) returns ...
        listBoxEdificio contents as cell array
130 %         contents{get(hObject,'Value')} returns selected item from ...
        listBoxEdificio
131
132
133 % --- Executes during object creation, after setting all properties.
134 function listBoxEdificio_CreateFcn(hObject, eventdata, handles)
135 % hObject    handle to listBoxEdificio (see GCBO)
136 % eventdata  reserved - to be defined in a future version of MATLAB
137 % handles    empty - handles not created until after all ...
        CreateFcns called
138
139 % Hint: listbox controls usually have a white background on Windows.
140 %         See ISPC and COMPUTER.
141 if ispc && isequal(get(hObject,'BackgroundColor'), ...
        get(0,'defaultUicontrolBackgroundColor'))
142     set(hObject,'BackgroundColor','white');
143 end
144
145
146 % --- Executes on button press in pushbuttonInicializar.
147 function pushbuttonInicializar_Callback(hObject, eventdata, handles)
148 % hObject    handle to pushbuttonInicializar (see GCBO)
149 % eventdata  reserved - to be defined in a future version of MATLAB
150 % handles    structure with handles and user data (see GUIDATA)
151 global Consumos Edificios G Localidades municipios provincias Tred ...
        pro2muni;
152 load('Edificios.mat');
153 load('provincias.mat');
154 set(handles.popupmenuProvincias,'String',provincias);
155 set(handles.popupmenuEdificios,'String',Edificios);
156 set(handles.popupmenuMaterial,'String',{'Aluminio','Acero ...
        galvanizado'});
157 set(handles.popupmenuTamano,'String',{'Pequeno','Mediano','Grande'});
158 % set(gcf,'units','normalized','outerposition',[.3 .3 .7 .7])
159 set(gcf,'units','pixels','outerposition',[100 100 1270 650]);
160 clc;
161 imshow('LogoElectrica.png', 'Parent', handles.axesLogo);

```

```

162 % --- Executes on selection change in popupmenuProvincias.
163 function popupmenuProvincias_Callback(hObject, eventdata, handles)
164 % hObject      handle to popupmenuProvincias (see GCBO)
165 % eventdata    reserved - to be defined in a future version of MATLAB
166 % handles      structure with handles and user data (see GUIDATA)
167
168 % Hints: contents = cellstr(get(hObject,'String')) returns ...
        popupmenuProvincias contents as cell array
169 %           contents{get(hObject,'Value')} returns selected item from ...
        popupmenuProvincias
170
171 pro = get(hObject,'Value');
172 load('pro2muni.mat')
173 set(handles.popupmenuMunicipios,'Value',1);
174 set(handles.popupmenuMunicipios,'String',pro2muni{pro});
175
176 % --- Executes during object creation, after setting all properties.
177 function popupmenuProvincias_CreateFcn(hObject, eventdata, handles)
178 % hObject      handle to popupmenuProvincias (see GCBO)
179 % eventdata    reserved - to be defined in a future version of MATLAB
180 % handles      empty - handles not created until after all ...
        CreateFcns called
181
182 % Hint: popupmenu controls usually have a white background on Windows.
183 %           See ISPC and COMPUTER.
184 if ispc && isequal(get(hObject,'BackgroundColor'), ...
        get(0,'defaultUiControlBackgroundColor'))
185     set(hObject,'BackgroundColor','white');
186 end
187
188
189 % --- Executes on selection change in popupmenuMunicipios.
190 function popupmenuMunicipios_Callback(hObject, eventdata, handles)
191 % hObject      handle to popupmenuMunicipios (see GCBO)
192 % eventdata    reserved - to be defined in a future version of MATLAB
193 % handles      structure with handles and user data (see GUIDATA)
194
195 % Hints: contents = cellstr(get(hObject,'String')) returns ...
        popupmenuMunicipios contents as cell array

```

```

196 %         contents{get(hObject,'Value')} returns selected item from ...
           popupmenuMunicipios
197
198
199 % --- Executes during object creation, after setting all properties.
200 function popupmenuMunicipios_CreateFcn(hObject, eventdata, handles)
201 % hObject    handle to popupmenuMunicipios (see GCBO)
202 % eventdata  reserved - to be defined in a future version of MATLAB
203 % handles    empty - handles not created until after all ...
           CreateFcns called
204
205 % Hint: popupmenu controls usually have a white background on Windows.
206 %         See ISPC and COMPUTER.
207 if ispc && isequal(get(hObject,'BackgroundColor'), ...
           get(0,'defaultUicontrolBackgroundColor'))
208     set(hObject,'BackgroundColor','white');
209 end
210
211
212 % --- Executes on selection change in popupmenuEdificios.
213 function popupmenuEdificios_Callback(hObject, eventdata, handles)
214 % hObject    handle to popupmenuEdificios (see GCBO)
215 % eventdata  reserved - to be defined in a future version of MATLAB
216 % handles    structure with handles and user data (see GUIDATA)
217
218 % Hints: contents = cellstr(get(hObject,'String')) returns ...
           popupmenuEdificios contents as cell array
219 %         contents{get(hObject,'Value')} returns selected item from ...
           popupmenuEdificios
220 load('Consumos.mat');
221 load Edificios
222 e1 = get(handles.popupmenuEdificios,'Value');
223 set(handles.textn1,'String',sprintf('Cantidad de %s que ...
           corresponden al edificio tipo %s: ',Consumos{e1},Edificios{e1}));
224 if e1 == 17 % En caso de viviendas multifamiliares
225     set(handles.textn2,'Enable','on');
226     set(handles.editn2,'Enable','on');
227 else
228     set(handles.textn2,'Enable','off');

```

```
229     set(handles.editn2,'Enable','off');
230 end
231
232 % --- Executes during object creation, after setting all properties.
233 function popupmenuEdificios_CreateFcn(hObject, eventdata, handles)
234 % hObject    handle to popupmenuEdificios (see GCBO)
235 % eventdata  reserved - to be defined in a future version of MATLAB
236 % handles    empty - handles not created until after all ...
237             CreateFcns called
238
239 % Hint: popupmenu controls usually have a white background on Windows.
240 %       See ISPC and COMPUTER.
241 if ispc && isequal(get(hObject,'BackgroundColor'), ...
242     get(0,'defaultUiControlBackgroundColor'))
243     set(hObject,'BackgroundColor','white');
244 end
245
246 function editn1_Callback(hObject, eventdata, handles)
247 % hObject    handle to editn1 (see GCBO)
248 % eventdata  reserved - to be defined in a future version of MATLAB
249 % handles    structure with handles and user data (see GUIDATA)
250
251 % Hints: get(hObject,'String') returns contents of editn1 as text
252 %       str2double(get(hObject,'String')) returns contents of ...
253             editn1 as a double
254
255 % --- Executes during object creation, after setting all properties.
256 function editn1_CreateFcn(hObject, eventdata, handles)
257 % hObject    handle to editn1 (see GCBO)
258 % eventdata  reserved - to be defined in a future version of MATLAB
259 % handles    empty - handles not created until after all ...
260             CreateFcns called
261
262 % Hint: edit controls usually have a white background on Windows.
263 %       See ISPC and COMPUTER.
```

```
263 if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUiControlBackgroundColor'))
264     set(hObject,'BackgroundColor','white');
265 end
266
267
268 function editn2_Callback(hObject, eventdata, handles)
269 % hObject    handle to editn2 (see GCBO)
270 % eventdata  reserved - to be defined in a future version of MATLAB
271 % handles    structure with handles and user data (see GUIDATA)
272
273 % Hints: get(hObject,'String') returns contents of editn2 as text
274 %         str2double(get(hObject,'String')) returns contents of ...
           editn2 as a double
275
276
277 % --- Executes during object creation, after setting all properties.
278 function editn2_CreateFcn(hObject, eventdata, handles)
279 % hObject    handle to editn2 (see GCBO)
280 % eventdata  reserved - to be defined in a future version of MATLAB
281 % handles    empty - handles not created until after all ...
           CreateFcns called
282
283 % Hint: edit controls usually have a white background on Windows.
284 %         See ISPC and COMPUTER.
285 if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUiControlBackgroundColor'))
286     set(hObject,'BackgroundColor','white');
287 end
288
289
290 % --- Executes on selection change in popupmenuMaterial.
291 function popupmenuMaterial_Callback(hObject, eventdata, handles)
292 % hObject    handle to popupmenuMaterial (see GCBO)
293 % eventdata  reserved - to be defined in a future version of MATLAB
294 % handles    structure with handles and user data (see GUIDATA)
295
296 % Hints: contents = cellstr(get(hObject,'String')) returns ...
           popupmenuMaterial contents as cell array
```

```

297 %           contents{get(hObject,'Value')} returns selected item from ...
           popupmenuMaterial
298
299
300 % --- Executes during object creation, after setting all properties.
301 function popupmenuMaterial_CreateFcn(hObject, eventdata, handles)
302 % hObject       handle to popupmenuMaterial (see GCBO)
303 % eventdata     reserved - to be defined in a future version of MATLAB
304 % handles       empty - handles not created until after all ...
           CreateFcns called
305
306 % Hint: popupmenu controls usually have a white background on Windows.
307 %           See ISPC and COMPUTER.
308 if ispc && isequal(get(hObject,'BackgroundColor'), ...
           get(0,'defaultUiControlBackgroundColor'))
309     set(hObject,'BackgroundColor','white');
310 end
311
312
313 % --- Executes on selection change in popupmenuTamano.
314 function popupmenuTamano_Callback(hObject, eventdata, handles)
315 % hObject       handle to popupmenuTamano (see GCBO)
316 % eventdata     reserved - to be defined in a future version of MATLAB
317 % handles       structure with handles and user data (see GUIDATA)
318
319 % Hints: contents = cellstr(get(hObject,'String')) returns ...
           popupmenuTamano contents as cell array
320 %           contents{get(hObject,'Value')} returns selected item from ...
           popupmenuTamano
321
322
323 % --- Executes during object creation, after setting all properties.
324 function popupmenuTamano_CreateFcn(hObject, eventdata, handles)
325 % hObject       handle to popupmenuTamano (see GCBO)
326 % eventdata     reserved - to be defined in a future version of MATLAB
327 % handles       empty - handles not created until after all ...
           CreateFcns called
328
329 % Hint: popupmenu controls usually have a white background on Windows.

```

```

330 %         See ISPC and COMPUTER.
331 if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
332     set(hObject,'BackgroundColor','white');
333 end
334
335
336
337 function editTacum_Callback(hObject, eventdata, handles)
338 % hObject     handle to editTacum (see GCBO)
339 % eventdata   reserved - to be defined in a future version of MATLAB
340 % handles     structure with handles and user data (see GUIDATA)
341
342 % Hints: get(hObject,'String') returns contents of editTacum as text
343 %         str2double(get(hObject,'String')) returns contents of ...
    editTacum as a double
344
345
346 % --- Executes during object creation, after setting all properties.
347 function editTacum_CreateFcn(hObject, eventdata, handles)
348 % hObject     handle to editTacum (see GCBO)
349 % eventdata   reserved - to be defined in a future version of MATLAB
350 % handles     empty - handles not created until after all ...
    CreateFcns called
351
352 % Hint: edit controls usually have a white background on Windows.
353 %         See ISPC and COMPUTER.
354 if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
355     set(hObject,'BackgroundColor','white');
356 end
357
358
359
360 function editTacs_Callback(hObject, eventdata, handles)
361 % hObject     handle to editTacs (see GCBO)
362 % eventdata   reserved - to be defined in a future version of MATLAB
363 % handles     structure with handles and user data (see GUIDATA)
364

```

```

365 % Hints: get(hObject,'String') returns contents of editTacs as text
366 %           str2double(get(hObject,'String')) returns contents of ...
           editTacs as a double
367
368
369 % --- Executes during object creation, after setting all properties.
370 function editTacs_CreateFcn(hObject, eventdata, handles)
371 % hObject     handle to editTacs (see GCBO)
372 % eventdata   reserved - to be defined in a future version of MATLAB
373 % handles     empty - handles not created until after all ...
           CreateFcns called
374
375 % Hint: edit controls usually have a white background on Windows.
376 %           See ISPC and COMPUTER.
377 if ispc && isequal(get(hObject,'BackgroundColor'), ...
           get(0,'defaultUiControlBackgroundColor'))
378     set(hObject,'BackgroundColor','white');
379 end
380
381
382
383 function editRho_Callback(hObject, eventdata, handles)
384 % hObject     handle to editRho (see GCBO)
385 % eventdata   reserved - to be defined in a future version of MATLAB
386 % handles     structure with handles and user data (see GUIDATA)
387
388 % Hints: get(hObject,'String') returns contents of editRho as text
389 %           str2double(get(hObject,'String')) returns contents of ...
           editRho as a double
390
391
392 % --- Executes during object creation, after setting all properties.
393 function editRho_CreateFcn(hObject, eventdata, handles)
394 % hObject     handle to editRho (see GCBO)
395 % eventdata   reserved - to be defined in a future version of MATLAB
396 % handles     empty - handles not created until after all ...
           CreateFcns called
397
398 % Hint: edit controls usually have a white background on Windows.

```

```
399 %           See ISPC and COMPUTER.
400 if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
401     set(hObject,'BackgroundColor','white');
402 end
403
404
405
406 function editCp_Callback(hObject, eventdata, handles)
407 % hObject     handle to editCp (see GCBO)
408 % eventdata   reserved - to be defined in a future version of MATLAB
409 % handles     structure with handles and user data (see GUIDATA)
410
411 % Hints: get(hObject,'String') returns contents of editCp as text
412 %           str2double(get(hObject,'String')) returns contents of ...
    editCp as a double
413
414
415 % --- Executes during object creation, after setting all properties.
416 function editCp_CreateFcn(hObject, eventdata, handles)
417 % hObject     handle to editCp (see GCBO)
418 % eventdata   reserved - to be defined in a future version of MATLAB
419 % handles     empty - handles not created until after all ...
    CreateFcns called
420
421 % Hint: edit controls usually have a white background on Windows.
422 %           See ISPC and COMPUTER.
423 if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
424     set(hObject,'BackgroundColor','white');
425 end
426
427
428 % --- Executes on button press in pushbuttonCalcular.
429 function pushbuttonCalcular_Callback(hObject, eventdata, handles)
430 % hObject     handle to pushbuttonCalcular (see GCBO)
431 % eventdata   reserved - to be defined in a future version of MATLAB
432 % handles     structure with handles and user data (see GUIDATA)
433 set(hObject,'String','Calculando...');
```

```

434 load inicializar
435 clc
436 [n,m] = size(municipios);
437 e1 = get(handles.popupmenuEdificios,'Value');
438 n1 = str2double(get(handles.editn1,'String'));
439 n2 = str2double(get(handles.editn2,'String'));
440
441     if (e1 == 16 || e1 == 17)      % En caso de viviendas
442         n1 = dorm2pax(n1);      % Se emplea el numero de personas en ...
                                vez de dormitorios
443     %         fprintf('A %i dormitorios le corresponden %i personas ...
                                \n',nn,m);
444     end
445     demanda = n1*n2*G(e1);
446     if demanda < 50
447         msgbox('Para consumos menores de 50 l/dia no es necesario ...
                                aporte solar');
448     return
449 end
450
451
452 ind1 = get(handles.popupmenuProvincias,'Value');
453 ind2 = get(handles.popupmenuMunicipios,'Value');
454
455 muni = pro2muni{ind1};
456 muni = muni{ind2};
457 for i = 1:m
458     if strcmp(municipios(i).municipio,muni)    % Encuentro el ...
                                municipio
459         indx = i;
460         break
461     end
462 end
463 datosMunicipio = municipios(indx);
464 for i = 1:m
465     if strcmp(municipios(i).municipio,datosMunicipio.capital)    ...
                                % Encuentro el municipio
466         indx = i;
467         break

```

```

468         end
469     end
470     datosCapital = municipios(indx);
471     for i = 1:m
472         if strcmp(Tred(i).Municipio,datosMunicipio.capital) % ...
            Encuentro la capital
473             indx = i;
474             break
475         end
476     end
477     datosCapital.Tred = Tred(i).Temperatura;
478     % Se calcula la temperatura media del agua de la red segun el CTE
479     datosMunicipio.Tred = ...
        Tred2Tred(datosCapital.Tred,datosCapital.altura,datosMunicipio.altura);
480
481
482 %% Calculo de la demanda
483 contribucion = dem2contrib(demanda,datosMunicipio.PVGIS.zona);
484 Tacum = str2double(get(handles.editTacum,'String')); % C
485 Tacs = str2double(get(handles.editTacs,'String')); % C
486 rho = str2double(get(handles.editRho,'String')); % kg/L
487 cp = str2double(get(handles.editCp,'String')); % kJ/kg/C
488 tamano = get(handles.popupmenuTamano,'Value');
489 material = get(handles.popupmenuMaterial,'Value');
490 ndias = [31 28 31 30 31 30 31 31 30 31 30 31]';
491 Di = demanda.*((60 - datosMunicipio.Tred)./(Tacs - ...
    datosMunicipio.Tred));
492
493 %% Calculo de los colectores
494 % Colector seleccionado
495 colector = datosColector(material,tamano);
496 S0 = demanda/70; % S0 [m2] = demanda(L/dia)/70
497 n0 = round(S0/colector.Sc); % Primera estimacion del n de ...
    colectores
498 sencilla = get(handles.radiobuttonSencilla,'Value');
499 doble = ~sencilla;
500 if sencilla % Si la superficie del colector es sencilla, ...
    (tau*alpha)/(tau*alpha)n = 0.96
501     tatan = 0.96;

```

```

502 elseif doble      % Si es doble, 0.94
503     tatan = 0.94;
504 end
505 FrprimaFr = 0.95;    % Factor de correccion del conjunto ...
    captador-intercambiador. Se recomienda tomar el valor de 0.95
506 Frprimata = tatan*FrprimaFr*colector.rto;    % Factor ...
    Frprima*(tau*alpha). [Adimensional]
507 FrprimaUl = FrprimaFr*colector.k1;           % Factor Frprima*Ul. ...
    [W/m2/K]
508 %% Calculo de la cantidad de acumuladores
509     n = str2double(get(handles.editn,'String')); % L
510     if n == 0      % Si no se ha modificado el valor por defecto, ...
        que es cero, se calcula
511         n = n0;
512     end
513     set(handles.editn,'String',num2str(n));
514 Scn = colector.Sc*n;
515 %% Calculo del volumen del acumulador
516     V_AC = str2double(get(handles.editV,'String')); % L
517     if V_AC == 0    % Si no se ha modificado el valor por defecto, ...
        que es cero, se calcula
518         V_AC = round(70*Scn,-2);    % Volumen de acumulacion [L] = ...
            70*Scn [m2]. Depende de la Ta de utilizacion. Redondeo ...
            a multiplos de 100 L
519     end
520     set(handles.editV,'String',num2str(V_AC));
521
522 K1 = (V_AC*rho/(75*Scn))^-0.25;
523 R1 = ...
    datosMunicipio.PVGIS.Hopt./1000.*(1-datosMunicipio.PVGIS.PerdidasSombras); ...
    % Radiacion diaria media mensual incidente sobre la superficie ...
    de captacion por m2 [kWh/m2/dia]
524 Ea = Scn.*Frprimata.*R1.*ndias; % Energia absorbida por el ...
    conjunto de captadores [kWh/mes]
525
526 K2 = (11.6 + 1.18.*Tacs + 3.86.*datosMunicipio.Tred' - ...
    2.32.*datosMunicipio.PVGIS.Td) ./ ...
527     (100 - datosMunicipio.PVGIS.Td);

```

```

528 Qa = cp.*demanda.*rho.*ndias.*(Tacs - datosMunicipio.Tred')/3600; ...
    % Carga calorifica mensual [kWh/mes]
529 Ep = ...
    Scn.*FrprimaUl.*(100-datosMunicipio.PVGIS.Td).*ndias.*24./1000.*K1.*K2;
530
531 D1 = Ea./Qa;
532 D2 = Ep./Qa;
533
534 f = 1.029.*D1 - 0.065.*D2 - 0.245.*D1.^2 + 0.018.*D2.^2 + ...
    0.0215.*D1.^3;
535 Qu = f.*Qa;
536 coberturaSolarAnual = sum(Qu)/sum(Qa);
537 fCO2 = 0.311; % kg de CO2 por cada kWh util
538 %% Comprobamos
539 [output,textoComprobaciones] = comprobaciones(demanda*rho,Scn,V_AC,f);
540 textoResumen = sprintf('La contribucion solar minima para el ...
    municipio escogido es de %i %%\nLa cobertura solar anual es del ...
    %i %%\nLa energia ahorrada de combustibles fosiles es de %.0f ...
    kWh/ano\nSe evitan la emision de %.0f kg de CO2 al ano\nEl ...
    angulo de inclinacion optimo de los colectores es de ...
    %i\n',round(contribucion*100),round(coberturaSolarAnual*100),sum(Qu),sum(Qu)*fCO2);
541
542
543 %% Pintamos
544 axes(handles.axesBarras);
545 bar([Qu Qa]);
546 xlabel('mes');
547 ylabel('Q [kWh/mes]')
548 grid on
549 legend('Q_u','Q_a','location','south');
550 %% Escribimos
551 set(handles.textComprobaciones,'String',textoComprobaciones);
552 set(handles.textResumen,'String',textoResumen);
553 if all(output) % Si todo es OK
554     set(handles.textComprobaciones,'BackgroundColor','green');
555 else
556     set(handles.textComprobaciones,'BackgroundColor',[1 .3 .3]);
557 end
558 set(hObject,'String','Calcular');

```

```
559
560
561 function editn_Callback(hObject, eventdata, handles)
562 % hObject      handle to editn (see GCBO)
563 % eventdata    reserved - to be defined in a future version of MATLAB
564 % handles      structure with handles and user data (see GUIDATA)
565
566 % Hints: get(hObject,'String') returns contents of editn as text
567 %          str2double(get(hObject,'String')) returns contents of ...
           editn as a double
568
569
570 % --- Executes during object creation, after setting all properties.
571 function editn_CreateFcn(hObject, eventdata, handles)
572 % hObject      handle to editn (see GCBO)
573 % eventdata    reserved - to be defined in a future version of MATLAB
574 % handles      empty - handles not created until after all ...
           CreateFcns called
575
576 % Hint: edit controls usually have a white background on Windows.
577 %          See ISPC and COMPUTER.
578 if ispc && isequal(get(hObject,'BackgroundColor'), ...
           get(0,'defaultUiControlBackgroundColor'))
579     set(hObject,'BackgroundColor','white');
580 end
581
582
583
584 function editV_Callback(hObject, eventdata, handles)
585 % hObject      handle to editV (see GCBO)
586 % eventdata    reserved - to be defined in a future version of MATLAB
587 % handles      structure with handles and user data (see GUIDATA)
588
589 % Hints: get(hObject,'String') returns contents of editV as text
590 %          str2double(get(hObject,'String')) returns contents of ...
           editV as a double
591
592
593 % --- Executes during object creation, after setting all properties.
```

```
594 function editV_CreateFcn(hObject, eventdata, handles)
595 % hObject    handle to editV (see GCBO)
596 % eventdata  reserved - to be defined in a future version of MATLAB
597 % handles    empty - handles not created until after all ...
           CreateFcns called
598
599 % Hint: edit controls usually have a white background on Windows.
600 %         See ISPC and COMPUTER.
601 if ispc && isequal(get(hObject,'BackgroundColor'), ...
           get(0,'defaultUicontrolBackgroundColor'))
602     set(hObject,'BackgroundColor','white');
603 end
604
605
606 % --- Executes on button press in pushbuttonCatColectores.
607 function pushbuttonCatColectores_Callback(hObject, eventdata, handles)
608 % hObject    handle to pushbuttonCatColectores (see GCBO)
609 % eventdata  reserved - to be defined in a future version of MATLAB
610 % handles    structure with handles and user data (see GUIDATA)
611 open('Captadores_Fagor.pdf')
612
613 % --- Executes on button press in pushbuttonCatAcum.
614 function pushbuttonCatAcum_Callback(hObject, eventdata, handles)
615 % hObject    handle to pushbuttonCatAcum (see GCBO)
616 % eventdata  reserved - to be defined in a future version of MATLAB
617 % handles    structure with handles and user data (see GUIDATA)
618 open('Acumuladores_Fagor.pdf')
619
620 % --- Executes on button press in pushbuttonReset.
621 function pushbuttonReset_Callback(hObject, eventdata, handles)
622 % hObject    handle to pushbuttonReset (see GCBO)
623 % eventdata  reserved - to be defined in a future version of MATLAB
624 % handles    structure with handles and user data (see GUIDATA)
625 set(handles.editV,'String','0');
626 set(handles.editn,'String','0');
```


Bibliografía

- [1] KALOGIROU, SOTERIS A., *Solar Energy Engineering: Processes and Systems*, Academic Press, 1 Ed., 2009.
- [2] GARCÍA GUTIÉRREZ, M.A., *El proyecto de las instalaciones de climatización. Cumplimiento de normativa.*, Ingeniería Industrial Instalaciones 3, Jaén, 2013.
- [3] GERHARD STRYI-HIPP, *Renewable Heating and Cooling: Technologies and Applications*, Woodhead Publishing, 1ª Ed, 2015.
- [4] IOAN SARBU, CALIN SEBARCHIEVICI, *Solar Heating and Cooling Systems. Fundamentals, Experiments and Applications*, Academic Press, 1ª Ed, 2016.
- [5] R.Z. WANG, T.S. GE, *Advances in Solar Heating and Cooling*, Woodhead Publishing, 1ª Ed, 2016.
- [6] SANFORD A. KLEIN, WILLIAM A. BECKMAN, Y JOHN A. DUFFIE, *A design procedure for solar heating systems*, Solar energy, Vol. 18. pp. 113-127. Pergamon Press 1976.
- [7] SANFORD A. KLEIN, WILLIAM A. BECKMAN, Y JOHN A. DUFFIE, *A design procedure for solar heating systems*, Solar energy, Vol. 19. pp. 509-512. Pergamon Press 1977.
- [8] WILLIAM A. BECKMAN, SANFORD A. KLEIN, JOHN A. DUFFIE, *Solar heating design by the f-chart method*, John Wiley and sons, 1ª Ed, 1977.
- [9] SANFORD A. KLEIN, *Calculation of flat-plate collector utilizability*, Solar energy, Vol. 21. pp. 393-402. Pergamon Press 1978.

- [10] SANFORD A. KLEIN Y WILLIAM A. BECKMAN, *A general design method for closed-loop solar energy systems*, Solar energy, Vol. 22. pp. 269-282. Pergamon Press 1979.
- [11] INSTITUTO PARA LA DIVERSIFICACIÓN Y AHORRO DE LA ENERGÍA, *Pliego de Condiciones Técnicas de Instalaciones de Baja Temperatura*, IDAE, 1ª Ed, 2009.
- [12] INSTITUTO PARA LA DIVERSIFICACIÓN Y AHORRO DE LA ENERGÍA, *Manuales de energías renovables: energía solar térmica*, IDAE, 1ª Ed, 2006.
- [13] INSTITUTO PARA LA DIVERSIFICACIÓN Y AHORRO DE LA ENERGÍA *Factores de emisión de CO₂ y coeficientes de paso a energía primaria de diferentes fuentes de energía final consumidas en el sector edificios en España*. 2016
- [14] VÁZQUEZ MORENO, J. Y HERRANZ AGUILAR, J. C., *Números gordos en el proyecto de instalaciones*, Cinter Divulgación Técnica, S.L.L., 1ª Ed, 2012.
- [15] VERA CANDEAS, D., *Apuntes de la asignatura: Energías Alternativas*, Universidad de Jaén, 2017.
- [16] *Código Técnico de la Edificación. Documento básico de ahorro de energía*.
- [17] JOSÉ A. C. GONZÁLEZ, ROQUE C. PÉREZ, ANTONIO C. SANTOS, MANUEL-A.C. GIL, *Centrales de energías renovables. Generación eléctrica con energías renovables*, Pearson Prentice Hall, 1ª Ed. 2009.
- [18] KALOGIROU, S., *Solar thermal collectors and applications*, Prog. Energ. Combust. Sci. 30 (3), 231–295. 2004
- [19] DELYANNIS, A., *Solar stills provide island inhabitants with water*, Sun at Work 10 (1), 6–8. 1967
- [20] DELYANNIS, E., BELESSIOTIS, V., *The history of renewable energies for water desalination*, Desalination 128, 147–159. 2000
- [21] MEINEL, A.B., MEINEL, M.P., *Appl. Solar Energ.—An Introduction.*, Addison-Wesley Publishing Company, Reading, MA. 1976

- [22] MOUCHOT, A., *esultat des experiences faites en divers points de l'Algerie, pour l'emploi industrielle de la chaleur solaire*, C. R. Acad. Sci. 86, 1019–1021. 1878
- [23] MOUCHOT, A., *Utilization industrielle de la chaleur solaire*, C. R. Acad. Sci. 90, 1212–1213. 1880.
- [24] KREIDER, J.F., KREITH, F., *Solar Heating and Cooling*, McGraw-Hill, New York. 1977.
- [25] KREIDER, J.F., KREITH, F., *Principles of Solar Engineering*, McGraw-Hill, New York. 1978.
- [26] JORDAN, R.C., IBELE, W.E., *Mechanical energy from solar energy.*, Proceedings of the World Symposium on Applied Solar Energy, pp. 81–101. 1956
- [27] SERI, *Power from the Sun: Principles of High Temperature Solar Thermal Technology.*, 1987
- [28] KALOGIROU, S., *Solar water heating in Cyprus—current status of technology and problems.*, Renewable Energy (10), 107–112. 1997
- [29] TURKENBURG, W.C., *Renewable Energy Technologies, World Energy Assessment, Chapter 7*, UNDP, 2000
- [30] PHILIBERT, C., *The Present and Future Use of Solar Thermal Energy as a Primary Source of Energy.*, International Energy Agency, Paris, France. 2005
- [31] BUSINESS INTELLIGENCE FÁCIL *Listado de los pueblos de España con su longitud, latitud y altitud.* URL: <https://www.businessintelligence.info/assets/listado-longitud-latitud-municipios-espana.html>
- [32] PVGIS *Overview of PVGIS data sources and calculation methods.* URL: http://re.jrc.ec.europa.eu/pvg_static/methods.html