

GRADO EN INGENIERÍA EN TECNOLOGÍA
INDUSTRIAL
TRABAJO FIN DE GRADO

***<SOLUCIONES TÉCNICAS EN
FACHADAS DE EDIFICIOS MEDIANTE
APLICACIÓN DE LA FLUIDODINÁMICA
COMPUTACIONAL>***

Alumno/Alumna: <Aristondo, De Miguel, Ander>
Director/Directora: <Blanco, Ilzarbe, Jesus Mari>

Curso: <2017-2018>

Fecha: < 19, junio, 2018>



BILBOKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE BILBAO

AGRADECIMIENTOS

Me gustaría agradecer al grupo de investigación IT781-13 “*Sostenibilidad, Construcción y Materiales*” por todas las facilidades aportadas para llevar a cabo este proyecto. Más en concreto, me gustaría agradecer al investigador Jesús Mari Blanco por todo el apoyo y ayuda recibidos de su parte. Gracias por todo el tiempo que has invertido en mi proceso de aprendizaje y por todos los consejos y sugerencias, sin los cuales este proyecto no sería tan completo.

RESUMEN TRILINGÜE

En el presente proyecto se desarrolla una herramienta de software visual, sencilla e interactiva que permite modelar, a partir de sus aspectos constructivos y las condiciones climatológicas externas, el comportamiento térmico de una fachada de doble piel de chapa perforada. El ejecutable se desarrolla en el lenguaje Visual Basic.NET, utilizando el entorno de desarrollo integrado de Microsoft Visual Studio. El modelo matemático que emplea el ejecutable para modelar la fachada está basado en un modelo desarrollado y publicado por el grupo de investigación “Sostenibilidad, Construcción y Materiales”. Dicho grupo implementó su modelo en unas funciones de Matlab®, de las cuales se parte para el desarrollo del ejecutable.

Palabras clave: *ejecutable, fachada de doble piel, chapa perforada, interactiva*

In this project, a user-friendly, simple and interactive software application is developed. The application models the thermal behaviour of double skin sheet-metal façades based on its building facts and external environmental conditions. The programming language used for developing the application is Visual Basic.NET, and compiled using the Microsoft Visual Studio integrated development environment. The mathematical model applied by the application to model the façade is based on a model developed and published by the research group “Sostenibilidad, Construcción y Materiales”. The group implemented the model on some Matlab® scripts, which are the base of the development of this application.

Key words: *application, double skin façade, perforated sheet-metal, interactive*

Proiektu honetan software aplikazio simple eta interaktibo bat garatzen da. Aplikazioak, fatxadaren izaera konstruktibo- eta eguraldi baldintzetatik abiatuta, metal xafla zulatuko azal bikoitzeko fatxada baten portaera termikoa modelatzen du. Aplikazioaren garapena Visual Basic.NET programazio hizkuntza erabiliz burutzen da. Kodea konpilatzeko, Microsoft Visual Studio garapen ingurune integratua erabiltzen da. Fatxada modelatzeko erabiltzen den eredu matematikoa “Sostenibilidad, Construcción y Materiales” ikerkuntza taldeak garatu eta argitaraturiko ereduan dago oinarritua. Ikerkuntza talde horrek, garaturiko eredua Matlab® softwarearekin erabiltzeko bi funtzioetan idatzi zuen. Funtzio horiek dira aplikazio honen garapenerako oinarria.

Hitz gakoak: *aplikazio, azal bikoitzeko fatxada, metal xafla zulatua, interaktibo*

ÍNDICE

1.	INTRODUCCIÓN.....	1
2.	CONTEXTO.....	2
2.1.	Fachada ventilada de doble piel de chapa perforada.....	2
2.2.	Transferencia de calor en fachadas ventiladas de doble piel de chapa perforada.....	4
2.3.	Eficiencia energética.....	6
3.	OBJETIVOS Y ALCANCE	7
3.1.	Objetivos	7
3.2.	Alcance.....	7
4.	ANÁLISIS DE ALTERNATIVAS	9
5.	METODOLOGÍA	10
5.1.	El lenguaje VB.NET	10
5.2.	El modelo matemático	11
5.2.1.	Balance de calor en la chapa.....	12
5.2.2.	Balance de calor en el vidrio.....	15
5.3.	Estructura del ejecutable	16
5.3.1.	Interfaz gráfica.....	18
5.3.2.	Código escrito.....	34
5.4.	Algoritmos matemáticos.....	42
5.4.1.	Cálculo de las temperaturas de la chapa y el vidrio	42
5.4.2.	Cálculo de la demanda de HVAC	45
5.4.3.	Consumo y ahorro de iluminación, calefacción y aire acondicionado.....	47
6.	EJEMPLO DE APLICACIÓN.....	49
6.1.	Introducción de los datos.....	49
6.2.	Obtención de los resultados	54
6.2.1.	Resultados numéricos	55
6.2.2.	Representación gráfica de los resultados	58
7.	PROPUESTAS DE MEJORA.....	65
8.	PLANIFICACIÓN	66
8.1.	Definición de tareas	66
8.2.	Diagrama de Gantt	68
9.	CONCLUSIONES.....	69

ANEXO I: CÓDIGO DEL EJECUTABLE	70
FormPrincipal	70
FormGraficos.....	94
FormConsumoAhorro	98
FormUserDefMaterial	98
FormIrradTamb	99
modCalculos	100
modGraficos.....	113
modDatos.....	134
ANEXO II: EJEMPLO DE INFORME. Caso <i>Bilbao_verano</i>	142
BIBLIOGRAFÍA.....	144

ÍNDICE DE TABLAS

Tabla 1: Valoración de las alternativas	9
Tabla 2: Controles gráficos de VB.NET utilizados en el ejecutable	19
Tabla 3: Parámetros de entrada de la función <i>Calcular_tchapa_vidrio</i>	38
Tabla 4: Parámetros de entrada de la función <i>Calcular_Ilu_Heat_AC</i>	38
Tabla 5: Parámetros de entrada del procedimiento <i>Generar_grafico</i>	41
Tabla 6: Valor de los factores de forma del aire, el cielo y el suelo	47

ÍNDICE DE FIGURAS

Figura 1: Fachada de doble piel de chapa perforada	3
Figura 2: Edificio de la EUITI de Bilbao.....	4
Figura 3: Variación de la temperatura del aire de la cámara en su movimiento ascendente. (Imagen tomada de [6])	5
Figura 4: Características principales de VB.NET	11
Figura 5: Balance de calor en la chapa. (Imagen tomada de [4])	13
Figura 6: Balance de calor en el vidrio. (Imagen tomada de [4])	15
Figura 7: Ventana principal del ejecutable	17
Figura 8: Logo del grupo "Sostenibilidad, Construcción y Materiales"	20
Figura 9: Pestaña de proyecto	21
Figura 10: Pestaña de introducción de parámetros de la chapa.....	21
Figura 11: Pestaña de introducción de parámetros de la cámara	22
Figura 12: Pestaña de introducción de parámetros ambientales.....	23
Figura 13: Pestaña de gráficos	24
Figura 14: Controles <i>trackIrradiancia</i> , <i>trackPerfRate</i> y <i>trackTamb</i>	25
Figura 15: Controles <i>checkAutomatico</i> , <i>checkSinChapa</i> , <i>checkChapaMaciza</i> , <i>btnCalcular</i> y <i>btnConsumoAhorro</i>	26
Figura 16: Controles <i>pic30cm</i> , <i>lbtChapa</i> , <i>lbtVidrio</i> , <i>lbtGap</i> , <i>lbtHVAC</i> y <i>picHVAC</i>	27

Figura 17: FormConsumoAhorro.....	28
Figura 18: FormIrradTamb	29
Figura 19: FormUserDefMaterial.....	30
Figura 20: Gráfico de comparación simple.....	32
Figura 21: Gráfico de comparación doble	33
Figura 22: Ejemplo de MessageBox	34
Figura 23: Ejemplo de InputBox.....	34
Figura 24: Esquema del código	36
Figura 25: Intercambio de calor entre la fachada y el interior del edificio. (Imagen tomada de [4]).....	46
Figura 26: Cuadro de dialogo de introducción del nombre del proyecto.....	49
Figura 27: Resumen de proyectos	50
Figura 28: Datos de la chapa	51
Figura 29: Introducción manual de los parámetros del material de la chapa	51
Figura 30: Introducción de la anchura de la cámara	52
Figura 31: Introducción manual de las condiciones ambientales.....	52
Figura 32: Determinar la ubicación	53
Figura 33: Introducción de datos de clima veraniego en Bilbao	53
Figura 34: Ejemplo de datos de proyecto actualizados	54
Figura 35: Resultados en el resumen	55
Figura 36: Temperaturas coincidentes en el resumen y en el esquema constructivo ..	56
Figura 37: Temperaturas diferentes en el resumen y en el esquema constructivo	57
Figura 38: Consumo y ahorro anuales de Bilbao	58
Figura 39: Pestaña de Gráficos	59
Figura 40: Datos para graficar p vs HVAC en Bilbao_verano.....	60
Figura 41: Gráfico de p vs HVAC para <i>Bilbao_verano</i>	60
Figura 42: Gráfico de p vs HVAC para <i>Bilbao_invierno</i>	61
Figura 43: Ventana de gráficos para comparar materiales.....	62
Figura 44: Comparativa de materiales en la demanda de HVAC para <i>Bilbao_invierno</i> 63	
Figura 45: Comparativa de materiales en el consumo de aire acondicionado para Bilbao 30 cm.....	63
Figura 46: Diagrama de Gantt.....	68

1. INTRODUCCIÓN

En este documento se presenta el desarrollo de una aplicación de software que modela el comportamiento térmico de una fachada ventilada de doble piel de chapa perforada.

En primer lugar se pone la fachada ventilada en contexto, exponiendo su origen y creciente interés en su uso. Además, se explican a rasgos generales los mecanismos de transferencia de calor que tienen lugar en este tipo de construcciones. A continuación, se establecen los objetivos y el alcance de este proyecto.

Después, se realiza un análisis de las diferentes alternativas contempladas para llevar a cabo este proyecto, describiendo mediante una tabla de ponderaciones los pros y contras de cada una de las opciones. Una vez seleccionada la alternativa, se procede a describir en detalle el contenido del ejecutable, así como los pasos seguidos en su desarrollo y cada uno de sus componentes y funciones. Tras la descripción, se explica mediante un ejemplo como aplicar la solución desarrollada. Para ello se plantea un caso de estudio, a modo de enunciado, y se procede a detallar paso por paso el proceso a seguir a la hora de introducir los datos y de obtener los resultados.

Tras el ejemplo, se propone una serie de mejoras que pueden llevarse a cabo sobre el ejecutable para ampliar tanto sus capacidades como su versatilidad. Finalmente, se termina con una serie de conclusiones en las que se resumen los puntos más importantes de este proyecto.

2. CONTEXTO

En este apartado se pone en contexto la fachada ventilada de doble piel. Se comienza explicando su historia y motivación, así como el creciente interés de su uso. A continuación se procede a describir el fenómeno de transferencia de calor que se da en este tipo de fachadas. Por último, se sitúa la fachada de doble piel en el marco de la eficiencia energética en edificación.

2.1. Fachada ventilada de doble piel de chapa perforada

La idea de una fachada ventilada se remonta a principios del siglo XX, cuando el arquitecto franco-suizo Le Corbusier comenzó a experimentar con dichas fachadas. La idea original consistía en construir un muro de cristal que protegiera la fachada del edificio de las condiciones externas, la cual incluiría tubos por los que circularía aire caliente. En 1930, un estudio realizado por ingenieros estadounidenses determinó que el sistema ideado por Le Corbusier supondría un consumo de energía mayor que cualquier sistema convencional de ventilación. Estudios posteriores determinaron que la idea podría mejorar la eficiencia energética de los edificios si la radiación solar es tomada en consideración. A día de hoy, las fachadas ventiladas están recibiendo un renovado interés debido a la creciente importancia del diseño eficiente en edificación [1].

La fachada ventilada de doble piel es un sistema constructivo de cerramiento exterior constituido por dos fachadas, una exterior y otra interior, separadas por una cámara por la que se permite la circulación de aire [1]. Estos sistemas son utilizados, generalmente, para mejorar la eficiencia energética de edificios cuyas fachadas poseen un alto porcentaje de vidrio en su superficie [2]. No obstante, juegan también un papel importante tanto a nivel decorativo, por dotar al edificio de una mejor presencia; así como a nivel de aislamiento acústico. Es por ello que las fachadas de doble piel son utilizadas en construcciones expuestas a un alto nivel de contaminación acústica como, por ejemplo, el ruido del tráfico [3].

Se denomina fachada ventilada de doble piel de chapa perforada a aquellas fachadas de doble piel cuya capa externa es, precisamente, chapa metálica perforada (ver [Figura 1](#)). En ellas, se utilizan materiales metálicos por su apariencia atractiva, así como por su durabilidad y resistencia [4]. La fachada exterior suele estar constituida por placas unidas mediante juntas que permiten su dilatación [1].



Figura 1: Fachada de doble piel de chapa perforada

Una de las características fundamentales de las fachadas ventiladas de chapa perforada es el grado de perforación, el cual se define como el área perforada entre el área total. El grado de perforación regula dos aspectos térmicos fundamentales de la fachada. Por un lado, regula la cantidad de radiación solar que incide de forma directa sobre la pared interior: cuanto mayor sea el grado de perforación, mayor será dicha cantidad. Por otro lado, regula el nivel de ventilación que tiene lugar en la cámara de aire: un mayor grado de perforación implica mayor ventilación en la cavidad [1].

Por último, debe tenerse en cuenta el efecto que el grado de perforación tiene sobre la visibilidad del edificio. Desde el exterior, la chapa perforada es lo primero que percibe el observador, por lo que debe cumplir su función decorativa [1]. En este sentido, una posibilidad interesante es la de dibujar formas sobre la fachada variando el grado de perforación. La [Figura 2](#) muestra un ejemplo de este efecto, que es la fachada de la Escuela Universitaria de Ingeniería Técnica Industrial de Bilbao (EUITI). Como bien se puede observar, el logo de la Universidad del País Vasco (UPV) aparece impreso en la fachada. Esto se consigue variando el grado de perforación, que es del 50% en toda la fachada a excepción del área de los logos, en la cual es del 35%. Desde el interior, la visibilidad es factible si el grado de perforación no es muy bajo.



Figura 2: Edificio de la EUITI de Bilbao

2.2. Transferencia de calor en fachadas ventiladas de doble piel de chapa perforada

El intercambio de calor en fachadas de doble piel de chapa perforada tiene lugar combinando los tres mecanismos de transferencia de calor: conducción, convección y radiación. La radiación solar incidente se distribuye entre chapa y pared de forma proporcional al grado de perforación [1]. Si bien es cierto que tanto la chapa como la pared se calientan debido a este efecto, la simple presencia de la chapa disminuye la cantidad de radiación que incide directamente sobre la pared interna, lo cual hace que su temperatura no se eleve tanto como lo haría en caso de estar completamente expuesta al sol. Además, en situaciones de escasa radiación solar, la chapa actúa como barrera para el calor que escapa de la pared interna, impidiendo que se enfríe tanto como lo haría si estuviera directamente expuesta al ambiente.

Debido a la acción del sol, las temperaturas de chapa y pared son superiores a las del ambiente, de forma que éstas transmiten calor al aire de la cámara. En la superficie tanto de la chapa como de la pared, se genera una capa delgada de aire más caliente que el del ambiente. La densidad del aire en dicha capa es más baja que la del aire

ambiente, puesto que a presión constante la densidad de un gas es inversamente proporcional a su temperatura [5]. Esta diferencia de densidad provoca que, de forma natural, el aire caliente suba por la cavidad generando una corriente ascendente similar a la que tiene lugar en las chimeneas (ver Figura 3). A medida que el aire caliente sube, el hueco que deja va siendo reemplazado por aire a temperatura ambiente (más frío). Este fenómeno se conoce como corriente de convección natural, y la transferencia de calor que se mejora como resultado de la misma se llama transferencia de calor por convección natural [5].

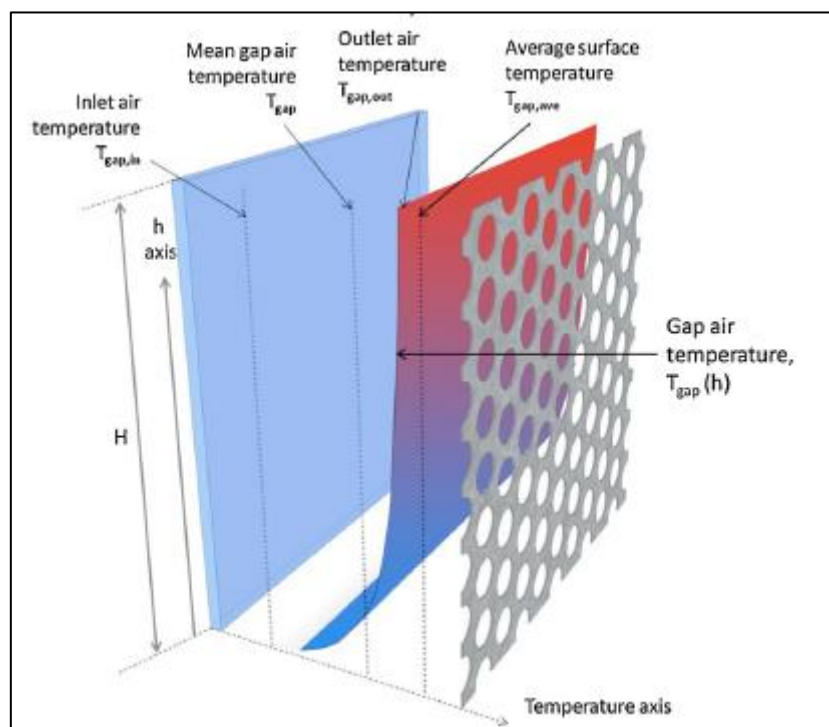


Figura 3: Variación de la temperatura del aire de la cámara en su movimiento ascendente. (Imagen tomada de [6])

Las perforaciones de la chapa permiten la entrada de aire externo a la cámara. En días soleados, al ser la temperatura de la chapa sensiblemente superior a la del ambiente, el aire se calienta al atravesar los agujeros. En este sentido, la chapa actúa de forma análoga a la resistencia de un calentador, transfiriendo parte del calor que absorbe del sol al aire [1]. A pesar de ello, al evitar que la radiación solar incida directamente sobre la pared del edificio, la chapa protege al edificio del calor solar.

El intercambio de calor con el interior del edificio tiene lugar por conducción a través de las paredes y ventanas. En días calurosos, el interior del edificio se encontrará a una temperatura inferior a la de la fachada y absorberá parte de su calor. En días fríos,

por el contrario, la temperatura interior será superior a la del ambiente; por lo parte del calor interno será cedido al exterior.

Por último, el edificio intercambia calor en forma de radiación con todo su entorno. La radiación que llega directamente del sol es en parte absorbida tanto por la chapa como por la pared interior, y en parte reflejada. La cantidad total absorbida y reflejada es función de los materiales y del color de la chapa, así como de la pared interna y de las ventanas. Por otra parte, el edificio emite calor en forma de radiación de onda larga por el simple hecho de estar a una temperatura superior al cero absoluto. Dicho calor es intercambiado con el ambiente, el suelo, el cielo y cualquier cuerpo u objeto del entorno del mismo que se halle a una temperatura diferente.

2.3. Eficiencia energética

En la sociedad de hoy en día, la sostenibilidad medioambiental está adquiriendo una gran relevancia a nivel político y económico [4]. La tendencia actual es la búsqueda de un equilibrio que permita niveles de desarrollo adecuados sin llegar a derrochar los recursos disponibles. En consecuencia, en el sector de la construcción se realizan cada vez más análisis de impacto ambiental de los edificios; así como estudios de cómo cada edificio depende de su entorno, buscando adaptarlos lo mejor posible de modo que su consumo sea mínimo. En este contexto, la Comunidad Económica Europea publicó la directiva 2002/91/EC [7] con el objetivo de establecer pautas para promover la eficiencia energética en el diseño de edificios. En España, dicha directiva se tradujo en el Real Decreto 1027/2007 [8], convirtiéndose éste en elemento clave en la Regulación de Instalaciones Térmicas en Edificación [4].

Para poder cuantificar la eficiencia energética de un edificio es fundamental realizar estudios que determinen el ahorro energético de las diferentes soluciones constructivas. Existen numerosos y detallados estudios energéticos destinados a caracterizar el rendimiento energético de fachadas ventiladas pero, hasta la fecha, son pocos los que estudian el comportamiento de las fachadas ventiladas de chapa perforada [4].

3. OBJETIVOS Y ALCANCE

En este apartado se procede a establecer los objetivos del proyecto y su alcance. Se comienza con una descripción de los objetivos del proyecto, dividiéndolos en objetivos principales y secundarios. A continuación, se expone el alcance del proyecto.

3.1. Objetivos

El objetivo principal de este proyecto es, partiendo del modelo matemático descrito en [4], desarrollar una herramienta de software intuitiva, visual y sencilla para el análisis del comportamiento energético de edificios con fachada de doble piel de chapa perforada. Dicho software debe ser independiente de cualquier plataforma comercial durante su ejecución, de forma que pueda ejecutarse cualquier ordenador sin ningún requerimiento previo en especial. Además, el software se debe desarrollar a modo de ejecutable, de forma que no sea necesaria su instalación para poder utilizarlo.

El ejecutable debe ser una herramienta que permita, a personas con escasos o nulos conocimientos de ingeniería térmica y transmisión de calor, la comprensión del efecto que distintos aspectos constructivos de las fachadas ventiladas de chapa perforada tienen sobre la demanda y el consumo energético de los edificios de una forma visual. En este sentido, puede ser utilizado para justificar elecciones de material o ratio de perforación en la chapa.

Además, se pretende que el ejecutable pueda ser una herramienta de apoyo en el ámbito académico, proporcionando a estudiantes de grado o postgrado una herramienta de cálculo sencilla y funcional. De esta manera, se facilita a dichos estudiantes la comprensión de los fenómenos de intercambio de calor sin necesidad de recurrir a herramientas de software comerciales ni de resolver complejas ecuaciones de balance de calor.

Junto con los objetivos principales, se definen una serie de objetivos intermedios o secundarios, que son los que se citan a continuación:

- Comprensión, interpretación y análisis de los fenómenos de transferencia de calor que tienen lugar en fachadas ventiladas de chapa perforada.
- Implementación de un algoritmo matemático que permita la resolución de sistemas no lineales de ecuaciones.
- Desarrollo de una interfaz gráfica atractiva y visual.

3.2. Alcance

Este es un proyecto de desarrollo de una aplicación de software dividido en etapas bien definidas, que son las que se exponen a continuación.

En primer lugar, se realizará un análisis de las posibles alternativas para el desarrollo de la aplicación. Se hará una comparativa técnica y económica del coste, ventajas y desventajas de cada una de ellas y se optará por utilizar el lenguaje de programación Visual Basic .NET (en adelante, VB.NET).

A continuación, se procederá a la documentación y formación en el uso de dicho lenguaje, así como del entorno de desarrollo necesario. Para ello se contará con un manual de aprendizaje de VB.NET ([9]) y con la documentación online facilitada por Microsoft en su página web.

Paralelamente, se escribirá una función que permita la resolución del sistema de ecuaciones descrito en [4]. Dicha función aplicará el algoritmo de Newton-Raphson para la resolución de sistemas de ecuaciones no lineales, y será validada comparando los resultados obtenidos con los correspondientes a las funciones de Matlab[®] desarrolladas por *Jesús Mari et al.* en [4]. Todo este trabajo previo a la implementación de dicha función en VB.NET será realizado en Matlab[®] y Mathematica[®].

Después, se procederá al diseño de la interfaz de la aplicación. Será una interfaz gráfica de usuario (en adelante GUI¹) creada a partir de los múltiples controles gráficos proporcionados por VB.NET. Dicha interfaz permitirá al usuario introducir los parámetros ambientales y constructivos que serán utilizados por la aplicación en los cálculos.

Una vez diseñada la interfaz, se procederá a escribir el código del programa. Partiendo de los parámetros de entrada definidos por el usuario, el código realizará las siguientes operaciones:

- Cálculo de las temperaturas medias de la chapa, el vidrio y la cámara de la fachada.
- Cálculo del consumo de HVAC por unidad de superficie en la fachada.
- Estimación la demanda de iluminación, calefacción y aire acondicionado de un edificio bajo las condiciones especificadas.
- Representación gráfica de los resultados obtenidos.
- Creación de un informe, basado en una plantilla, en el cual se resuman los resultados generados junto con los correspondientes parámetros introducidos.

El código también asignará a cada uno de los controles gráficos una serie de acciones para permitir al usuario interactuar con la aplicación.

Por último, se validará el funcionamiento del ejecutable comparando los resultados obtenidos con los que se obtuvieron inicialmente en Matlab[®]. Además, se detectarán y corregirán todos los posibles errores en tiempo de ejecución.

¹ Graphical User Interface

4. ANÁLISIS DE ALTERNATIVAS

En este apartado se analizan y comparan las distintas alternativas contempladas para el desarrollo del ejecutable, y se selecciona la que mejor se ajusta a las características del proyecto.

Para realizar la comparativa, se ha utilizado la [Tabla 1](#). Las columnas de la tabla se corresponden con cada una de las alternativas contempladas, a excepción de la primera, en la cual se asigna un peso a cada una de las filas. Por su parte, cada una de las filas representa los aspectos que han sido valorados a la hora de escoger la alternativa que finalmente será utilizada en este proyecto.

Cada una de las opciones es valorada con un número entre 0 y 10, siendo el 0 la valoración más baja y el 10 la más alta. Una vez valorados todos los requisitos, se realizará una suma ponderada en cada columna, considerando los pesos establecidos en la primera columna.

	Peso	NI LabVIEW	Java	VB.NET	Visual C# .NET
Coste económico	60%	0	10	10	10
Curva de aprendizaje	20%	7	5	9	8
Calidad de la solución	20%	10	9	7	7
TOTAL		3,4	8,8	9,2	9,0

Tabla 1: Valoración de las alternativas

En vista de los resultados, se opta por utilizar VB.NET como lenguaje de desarrollo del ejecutable. El entorno de desarrollo en el que funciona VB.NET es el entorno de desarrollo integrado (en adelante IDE²) de Microsoft Visual Studio, el cual es completo y gratuito.

Las aplicaciones desarrolladas con VB.NET no poseen la versatilidad de lenguajes como Java o software más especializado como LabVIEW, ya que solo pueden ser ejecutadas en ordenadores con el sistema operativo de Windows. No obstante, VB.NET proporciona múltiples controles gráficos sencillos a la par que útiles y, sobre todo, tiene una curva de aprendizaje corta que permitirá, en el tiempo que dura el proyecto, alcanzar un mayor nivel de desarrollo del ejecutable.

² Integrated Development Environment

5. METODOLOGÍA

En este apartado se procede a detallar el proceso de diseño de la aplicación de software, así como su estructura. Se comienza con un breve preámbulo en el que se introducen el lenguaje de programación VB.NET y el entorno de desarrollo utilizado en este proyecto. A continuación, se describe el modelo matemático empleado para modelizar el comportamiento de la fachada de doble piel de chapa perforada. Después se expone la estructura interna del ejecutable, la cual se divide en dos partes: la interfaz gráfica y el código escrito. Por último se realiza una descripción del algoritmo de cálculo empleado por el ejecutable.

5.1. El lenguaje VB.NET

Antes de la existencia de Visual Basic existía otro lenguaje de programación llamado BASIC, que fue creado en 1964 por Thomas Eugene Kurtz y John George Kemeny. Este lenguaje se creó para que a los estudiantes de las universidades les resultara más sencillo aprender a programar. Con el paso del tiempo, el programa fue evolucionando y ganando popularidad en su uso. Con la irrupción de los sistemas operativos con interfaz gráfica, se crearon nuevos lenguajes de programación que utilizaban la sintaxis de BASIC, entre ellos Microsoft Visual Basic [10].

La versión más popular y extendida del lenguaje era Visual Basic 6. No obstante, dicha versión presentaba una serie de desventajas que fueron corregidas con la aparición, en el año 2002, del .NET Framework. Las características principales de VB.NET son:

- Separación de la creación de la interfaz gráfica y el código.
- Una barra de herramientas con los controles necesarios para la creación de formularios³.
- Un explorador de proyectos que permite visualizar y acceder a todos los formularios, controles y elementos que componen el proyecto⁴.

Todas estas características están señaladas en la [Figura 4](#). En dicha figura se puede también observar el aspecto inicial que presenta una aplicación vacía de VB.NET. Al crear un nuevo proyecto, el IDE genera un formulario vacío sobre el que posteriormente se añaden controles desde la barra de herramientas. Cada vez que se añade un nuevo control, éste se ve reflejado en el explorador de proyectos.

³ Objeto que contiene los controles gráficos del ejecutable. A nivel de usuario se conocen como la “ventana” del programa.

⁴ Colección de archivos que se pueden compilar para crear un programa [9].

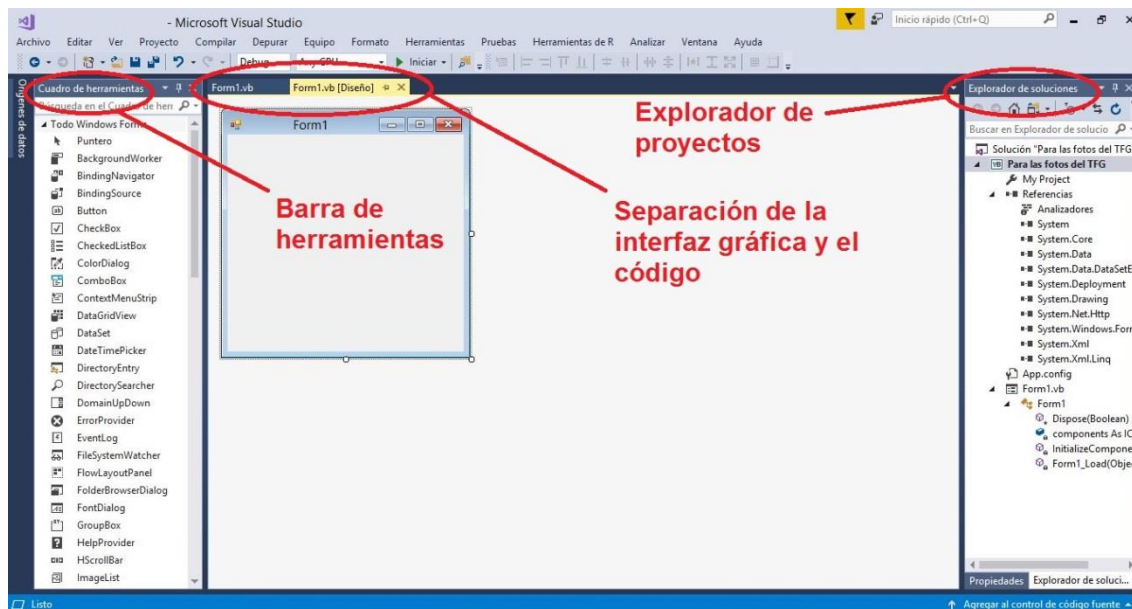


Figura 4: Características principales de VB.NET

El entorno de desarrollo utilizado para la creación del ejecutable es el IDE Microsoft Visual Studio 2017, el cual es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web, aplicaciones de escritorio y aplicaciones móviles. Entre otros, los lenguajes que ofrece para el desarrollo de aplicaciones de escritorio son los populares Visual Basic, Visual C# y Visual C++, para los cuales el IDE habilita el uso compartido de herramientas, facilitando así la creación de soluciones en distintos lenguajes [11].

El lenguaje VB.NET es un lenguaje de programación orientado a objetos y, como tal, puede decirse que todo lo que se utiliza en VB.NET es un objeto: los formularios, los controles que contienen dichos formularios, etcétera.

Un objeto es una estructura de programación que engloba datos y funcionalidades en una misma unidad y cuyo acceso se realiza a través de las interfaces de dicha estructura: propiedades, eventos y métodos. Las propiedades de un objeto son los atributos que se utilizan para especificar y devolver el estado de dicho objeto [9].

5.2. El modelo matemático

En ese apartado se describe el modelo matemático utilizado para modelar el intercambio de calor en la fachada ventilada. Se comentan y exponen las ecuaciones que gobiernan la transferencia de calor entre la fachada y el exterior, así como entre los distintos elementos que componen la propia fachada. Todas las ecuaciones e imágenes que se exponen en este apartado, así como el modelo matemático en sí han sido tomadas de [4].

El modelo para el estudio de las fachadas de doble piel de chapa perforada aplica un sistema de ecuaciones no lineales para estimar las temperaturas de la chapa y el vidrio. Los resultados de dicho sistema permiten, mediante balances de calor, calcular el calor que debe aportar o extraer el sistema de acondicionamiento térmico para mantener una temperatura interna de confort (ver apartado 5.4.3).

Como parámetros de entrada, se introducen variables climáticas como el nivel de irradiancia solar y la temperatura ambiente. Además, se introducen también características asociadas a factores constructivos de la fachada: propiedades térmicas de los materiales del vidrio y la chapa y el ratio de perforación.

Como hipótesis simplificadoras se considera que los coeficientes de convección son constantes y dependen de las temperaturas de la fachada y el ambiente. Se considera también que la temperatura de la cámara es constante a lo ancho. Se desprecia el intercambio de calor debido a la conducción, por ser la chapa muy fina y de materiales con alta conductividad térmica. Además, se considera también que la temperatura en todo el grosor del vidrio permanece constante.

El conjunto de ecuaciones se corresponde con los balances de calor de la chapa y el vidrio, que resulta en un sistema de dos ecuaciones no lineales que dependen de las temperaturas de la chapa y el vidrio, respectivamente.

5.2.1. Balance de calor en la chapa

Para realizar el balance de calor en la chapa, se consideran los flujos de calor que se muestran en la [Figura 5](#). En ella, se analizan por separado el intercambio de calor por convección (CO), por radiación de onda corta (SW) y por radiación de onda larga (LW). Esta notación se mantiene a lo largo de todo el desarrollo matemático de las ecuaciones. El significado del resto de los subíndices se ve reflejado en la parte inferior de la [Figura 5](#).

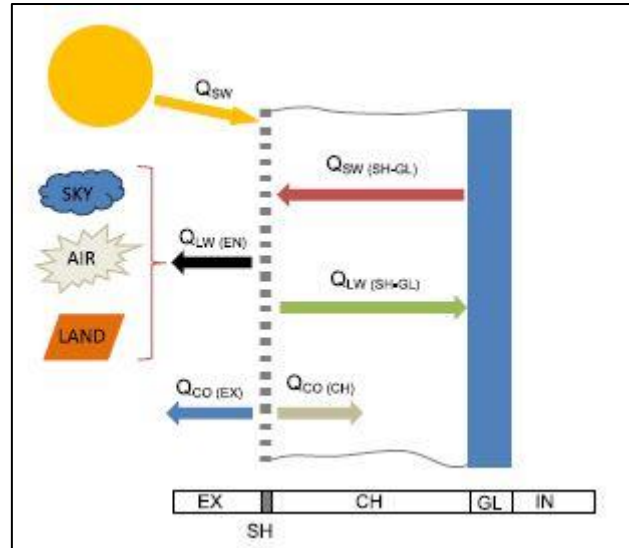


Figura 5: Balance de calor en la chapa. (Imagen tomada de [4])

La ecuación de balance de calor en la chapa⁵ que se muestra en la Figura 5 es la que se muestra a continuación.

$$Q_{SW} - Q_{LW(EN)} + Q_{SW(SH-GL)} - Q_{LW(SH-GL)} - Q_{CO} = 0$$

A continuación se obtiene el valor de cada uno de los términos de dicha ecuación, en función de las temperaturas del vidrio y de la chapa.

En primer lugar, el calor absorbido de radiación solar directa (de onda corta) es

$$Q_{SW} = I_{r_{SOL}} \cdot \alpha_{SOL(SH)} \cdot (1 - p)$$

El calor intercambiado por radiación de onda larga entre la chapa y los alrededores (suelo, aire y atmósfera) es

$$Q_{LW(EN)} = (Q_{LW(LA)} + Q_{LW(AI)} + Q_{LW(SK)}) \cdot (1 - p)$$

Donde

$$Q_{LW(LA)} = \varepsilon_{SH} \cdot \sigma \cdot F_{SH-LA} \cdot (T_{SH}^4 - T_{LA}^4)$$

$$Q_{LW(AI)} = \varepsilon_{SH} \cdot \sigma \cdot F_{SH-AI} \cdot (T_{SH}^4 - T_{AI}^4)$$

$$Q_{LW(SK)} = \varepsilon_{SH} \cdot \sigma \cdot F_{SH-SK} \cdot (T_{SH}^4 - T_{SK}^4)$$

⁵ El subíndice SH se corresponde con *chapa* y GL se corresponde con *vidrio*.

El calor intercambiado por radiación de onda corta entre la chapa y el vidrio es

$$Q_{SW(SH-GL)} = p \cdot I r_{SOL} \cdot \alpha_{SOL_{SH}} \cdot \left(\sum_{n=1}^{\infty} r_{GL}^n \cdot [(1-p) \cdot r_{SH}]^{n-1} \right)$$

Este término incluye el calor por radiación reflejado por el vidrio y absorbido por la chapa en sus infinitos rebotes. El intercambio de calor por radiación de onda larga entre la chapa y el vidrio debido a sus respectivas capacidades radiantes es

$$Q_{LW(SH-GL)} = \frac{\sigma \cdot (1-p) \cdot (T_{SH}^4 - T_{GL}^4)}{\frac{1}{\varepsilon_{SH}} + \frac{1}{\varepsilon_{GL}} + 1}$$

El término de convección es la suma del calor liberado por la chapa a la cámara de aire y al exterior.

$$Q_{CO} = (Q_{CO(CH)} + Q_{CO(EX)}) \cdot (1-p)$$

Donde cada uno de los términos es

$$Q_{CO(CH)} = h_{CH} \cdot (T_{SH} - T_{CH})$$

$$Q_{CO(EX)} = h_{EX} \cdot (T_{SH} - T_{EX})$$

Los coeficientes de convección considerados en estas expresiones se definen como:

$$h_{CH} = 1.52 \cdot |T_{SH} - T_{CH}|^{\frac{1}{3}}$$

$$h_{EX} = 1.31 \cdot |T_{SH} - T_{EX}|^{\frac{1}{3}} + 9$$

Cabe destacar que el coeficiente h_{EX} no coincide con el utilizado en [4]. Esto se debe a que, durante el desarrollo del ejecutable, se ha considerado más precisa la expresión indicada en este apartado.

La temperatura de la cámara⁶ viene definida por la siguiente expresión, la cual está basada en las hipótesis simplificativas expuestas en la introducción de este apartado.

$$T_{CH} = \frac{\frac{T_{SH} + T_{GL}}{2} + T_{EX}}{2}$$

⁶ El subíndice SH se corresponde con la cámara.

5.2.2. Balance de calor en el vidrio

La ecuación de balance de calor en el vidrio considera los flujos de calor de la [Figura 6](#). Se trata de un balance muy similar al desarrollado en el caso de la chapa, con ligeras modificaciones.

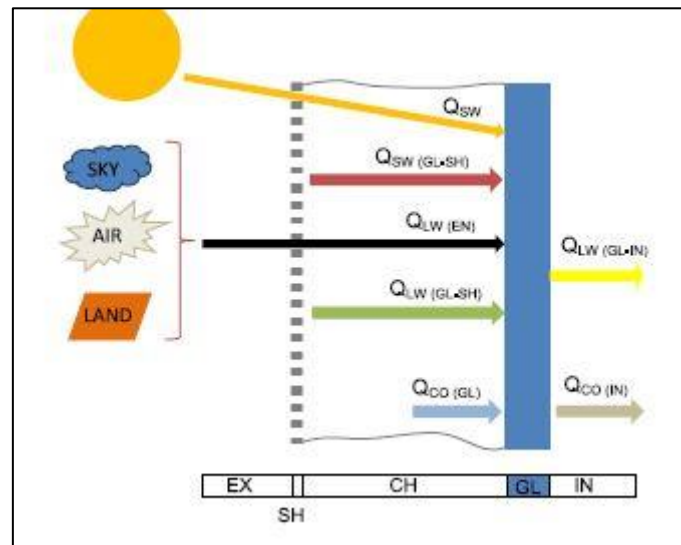


Figura 6: Balance de calor en el vidrio. (Imagen tomada de [4])

La ecuación de balance es

$$Q_{SW} + Q_{LW(EN)} + Q_{SW(GL-SH)} + Q_{LW(GL-SH)} - Q_{LW(GL-IN)} + Q_{CO} = 0$$

El calor absorbido por radiación solar directa y difusa (de onda corta) se obtiene mediante la expresión

$$Q_{SW} = I_{SOL} \cdot \alpha_{SOL(GL)} \cdot p$$

Por su parte, el calor intercambiado por el vidrio con el entorno (suelo, aire y atmósfera) se calcula sumando la contribución de cada uno de los elementos que componen el entorno

$$Q_{LW(EN)} = (Q_{LW(LA)} + Q_{LW(AI)} + Q_{LW(SK)}) \cdot p \cdot \tau_{GL}$$

Siendo cada uno de los términos idéntico a los considerados en el balance de la chapa, con la única diferencia de que deben considerarse la emisividad y temperatura del vidrio, en lugar de los de la chapa.

El calor de radiación de onda corta intercambiado por la chapa y el vidrio, que es la parte de la radiación directa que llega a cada uno de ellos y, tras rebotar, es finalmente absorbida es

$$Q_{SW(GL-SH)} = p \cdot I_{rSOL} \cdot \alpha_{SOL(GL)} \cdot \left(\sum_{n=1}^{\infty} r_{GL}^n \cdot [(1-p) \cdot r_{SH}]^n \right)$$

El calor intercambiado por radiación de onda larga entre el vidrio y la chapa debido a su propia capacidad radiante es

$$Q_{LW(GL-SH)} = \frac{\sigma \cdot (1-p) \cdot (T_{SH}^4 - T_{GL}^4)}{\frac{1}{\varepsilon_{SH}} + \frac{1}{\varepsilon_{GL}} + 1}$$

El calor intercambiado por radiación de onda larga entre el vidrio y el interior se obtiene mediante la expresión

$$Q_{LW(GL-IN)} = \varepsilon_{GL} \cdot \sigma \cdot F_{GL-IN} \cdot (T_{GL}^4 - T_{IN}^4)$$

EL término de convección se calcula restando al calor absorbido de la cámara el calor liberado al interior

$$Q_{CO} = Q_{CO(GL)} - Q_{CO(IN)}$$

Siendo cada uno de dichos términos

$$Q_{CO(GL)} = h_{CH} \cdot (T_{CH} - T_{GL})$$

$$Q_{CO(IN)} = h_{IN} \cdot (T_{CH} - T_{IN})$$

Donde el coeficiente de convección h_{IN} se calcula de la siguiente manera

$$h_{IN} = 1.31 \cdot |T_{GL} - T_{IN}|^{\frac{1}{3}}$$

5.3. Estructura del ejecutable

En este apartado se describen cada una de las partes que, a nivel de desarrollo, componen el ejecutable o solución final. Como se ha mencionado en el apartado anterior, una de las características de VB.NET es que permite la separación entre la interfaz gráfica y el código, por lo que cada uno de ellos será analizado por separado.

Al abrir el ejecutable, se abre en pantalla la ventana principal del programa (ver [Figura 7](#)). Dicha ventana proporciona una interfaz accesible que responde a las acciones que el usuario realiza sobre ella. Los elementos que permiten la interacción son los

denominados controles gráficos y la ventana es el formulario que los contiene que, en este caso, es también el formulario principal del programa.

Cuando el usuario actúa sobre alguno de los controles gráficos, por ejemplo haciendo click sobre un botón, el programa genera una respuesta de manera automática. Siguiendo con el ejemplo, la respuesta del programa al click realizado viene determinada por el código asociado a la acción click sobre ese botón en concreto. Dichas acciones reciben el nombre de eventos, y serán tratadas con mayor detalle en el subapartado de código.

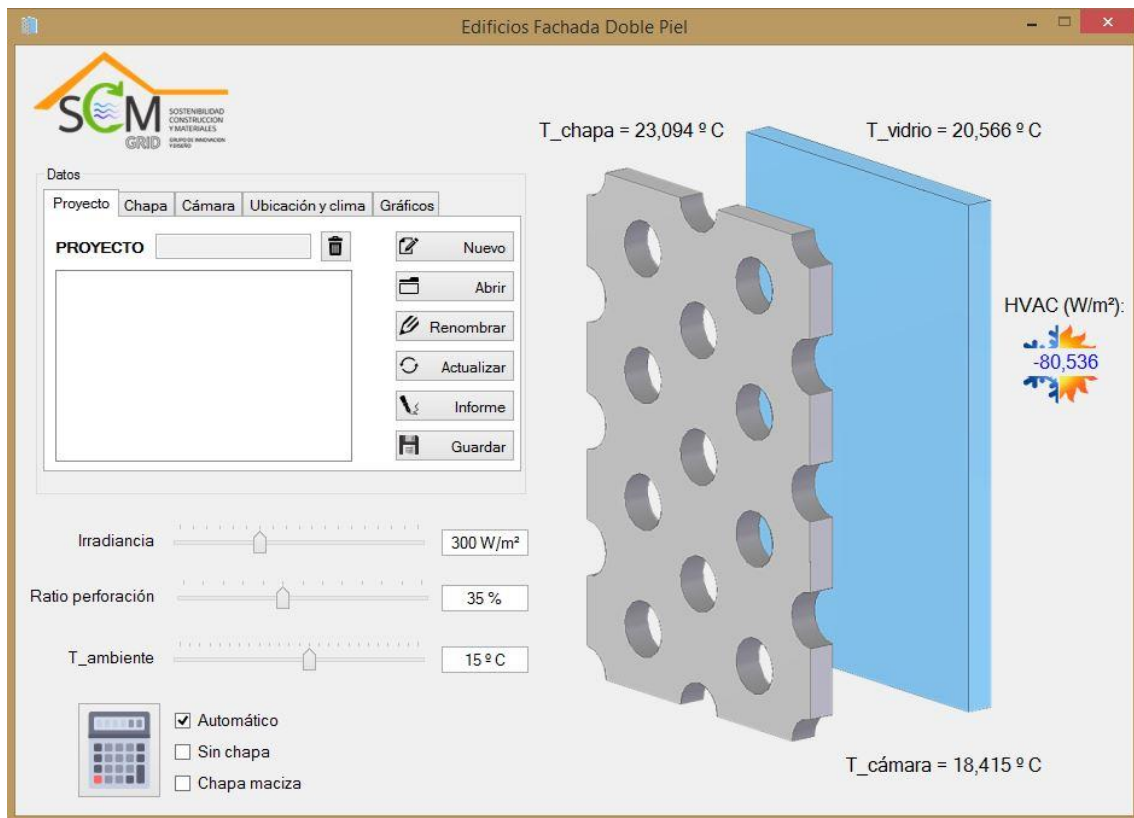


Figura 7: Ventana principal del ejecutable

La secuencia que se ha seguido en el desarrollo de cada una de las utilidades que ofrece el ejecutable es la siguiente:

- 1) En primer lugar, se selecciona el control más adecuado o con mejores prestaciones para llevar a cabo la acción que se desea implementar.
- 2) Después, se ubica el control en el formulario que vaya a contenerlo. Además de la ubicación, se determinan su tamaño, color, aspecto, visibilidad y demás propiedades gráficas.

- 3) Por último, se procede a la redacción del código asociado a cada uno de los eventos que deban producir una respuesta no nula por parte del programa.

5.3.1. Interfaz gráfica

En este subapartado se realiza un análisis exhaustivo de la interfaz gráfica del ejecutable, así como de todos los controles que en él intervienen. Se comienza con una descripción general de los tipos de controles gráficos disponibles en VB.NET, junto con su funcionalidad principal (ver [Tabla 2](#)). Después, se citan los formularios que contiene el ejecutable, explicando los controles presentes en cada uno de ellos y su función. Por último, se comentan las funciones MessageBox e InputBox, las cuales forman parte de la interfaz de usuario de forma indirecta.

Como se puede comprobar en la [Figura 4](#), la barra de herramientas de VB.NET posee una amplia gama de controles. Dichos controles presentan la ventaja de que son objetos ya programados que pueden utilizarse en el desarrollo de la interfaz de usuario sin necesidad de programarlos de cero. Además, poseen también la ventaja de que son controles habituales en aplicaciones de Windows, por lo que la mayoría de usuarios están familiarizados con ellos.

De forma general, los controles que incluye el ejecutable son los siguientes:

CONTROL	FUNCIONALIDAD
Button	Genera un evento cuando el usuario hace click en él.
ComboBox	Muestra un cuadro de texto editable con una lista desplegable de los valores permitidos.
Chart	Permite graficar datos.
CheckBox	Permite al usuario seleccionar o quitar la opción asociada.
GroupBox	Muestra un marco alrededor de un grupo de controles con un título opcional.
Label	Proporciona información en tiempo de ejecución o texto descriptivo para un control.
ListBox	Muestra una lista entre la que el usuario puede elegir elementos.
ListView	Muestra una colección de elementos
NumericUpDown	Muestra un único valor numérico que el usuario puede aumentar o reducir haciendo click en los botones de arriba y abajo del control.

PictureBox	Muestra una imagen.
RadioButton	Permite al usuario seleccionar una única opción entre un grupo de opciones cuando están emparejadas con otros RadioButtons.
TabControl	Controla y muestra al usuario una colección relacionada de pestañas que pueden contener controles y componentes.
TextBox	Permite al usuario especificar texto, así como funciones de edición de varias líneas y máscaras de caracteres para contraseñas.
TrackBar	Permite al usuario elegir entre un rango de valores que puede elegir desplazándose por una pequeña barra situada junto a otra barra.
TreeView	Muestra al usuario una colección jerárquica de elementos con etiqueta, que opcionalmente puede contener una imagen.

Tabla 2: Controles gráficos de VB.NET utilizados en el ejecutable

El ejecutable cuenta con cinco formularios, y cada uno de ellos posee sus propios controles y tiene una finalidad específica. A continuación se va a realizar un análisis exhaustivo de cada uno de ellos, exponiendo su razón de ser y explicando los controles que contiene y lo que cada uno de ellos hace.

5.3.1.1. FormPrincipal

FormPrincipal⁷ es el formulario principal del ejecutable (ver [Figura 7](#)). Su función es servir como interfaz principal del programa con el usuario, de forma que en él puedan introducirse todos los datos que el ejecutable requiere. Además, debe también mostrar los resultados de los cálculos o proporcionar una vía de acceso a ellos. El modo de acceder a él es, sencillamente, ejecutar la aplicación.

El formulario está dividido en dos mitades. La zona izquierda cuenta con un conjunto de controles destinados a la introducción de los parámetros de cálculo o ploteo de resultados, así como de utilidades varias. La zona derecha muestra un esquema constructivo de la fachada, junto con los resultados de temperaturas de chapa, vidrio y cámara y de HVAC.

Los controles que contiene el formulario principal son los siguientes:

picLogo

⁷ El nombre asignado a los objetos en el código del ejecutable está siempre compuesto por dos partes: un prefijo que indica el tipo de objeto del que se trata y un nombre que de una idea aproximada de su función. En este caso el prefijo Form indica que es un formulario y el nombre Principal indica que es el principal.



Se trata de un control de tipo PictureBox cuya función es mostrar en pantalla una imagen con el logo del grupo de investigación "Sostenibilidad Construcción y Materiales", el cual desarrolló el modelo de transferencia de calor en el que están basados los cálculos que realiza el ejecutable.



Figura 8: Logo del grupo "Sostenibilidad, Construcción y Materiales"

tabInputData

Se trata de un control de tipo TabControl cuya función es contener, de forma ordenada, una serie de pestañas que permitan al usuario configurar todos los parámetros de entrada para los cálculos. Está formado por una colección de cinco pestañas:

- 1) Proyecto: permite al usuario acceder, modificar y guardar los proyectos de trabajo, así como generar nuevos proyectos. La ventana, que se muestra en la [Figura 9](#), posee múltiples controles Button, un control TreeView, un control Label y un TextBox.

El control TreeView muestra una lista de elementos desplegados. Cada uno de ellos se corresponde con un proyecto y, al expandirlo, muestra un resumen de los datos que contiene cada proyecto. Cualquier modificación que el usuario desee guardar debe ser introducida por medio de los controles Button. Al hacerlo, dicha modificación se verá reflejada en el TreeView. El TextBox muestra el nombre del proyecto actual de trabajo y el Label simplemente indica que dicha casilla contiene el título del proyecto.



Figura 9: Pestaña de proyecto

- 2) Chapa: permite al usuario determinar aspectos constructivos de la chapa, en concreto, permite seleccionar el material y color de la chapa, así como el ratio de perforación. La selección de material y color se realiza por medio de cinco controles del tipo RadioButton, que permiten seleccionar entre aluminium anodized, clean metallic galvanized sheet, matte black painting, glossy white painting y user defined material. El último de estos controles abre un panel de introducción manual de datos. El ratio de perforación se introduce por medio de un control NumericUpDown.

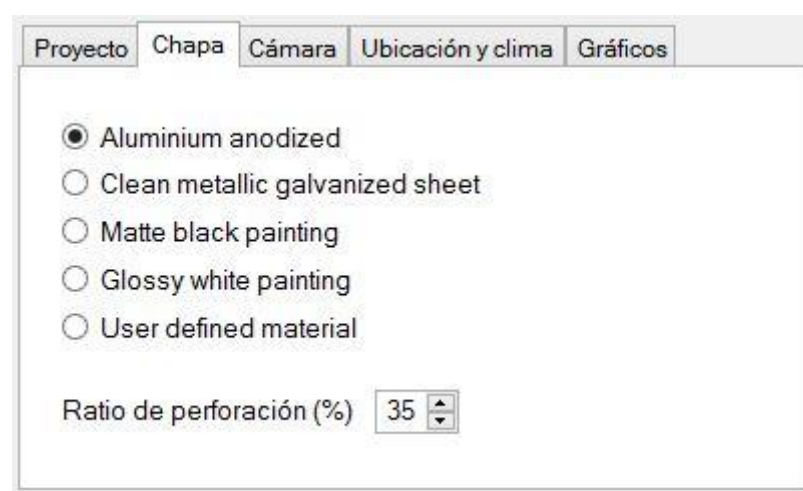


Figura 10: Pestaña de introducción de parámetros de la chapa

- 3) Cámara: permite al usuario especificar el ancho de la cámara, es decir, la separación entre la chapa y el vidrio. Para ello, cuenta con dos controles del tipo RadioButton: 5cm y 30cm. Además, contiene un control del tipo PictureBox que muestra una imagen del esquema constructivo del tipo de edificio modelado.



Figura 11: Pestaña de introducción de parámetros de la cámara

- 4) Ubicación y clima: permite al usuario especificar las condiciones climáticas (temperatura ambiente e irradiación solar) a las que se haya sometido el edificio. Para ello, cuenta con dos opciones: seleccionar la ubicación geográfica, mes del año y tipo de día, de forma que el propio programa genera automáticamente las condiciones de irradiación y temperatura ambiente; o introducir manualmente las condiciones que el usuario desee. La primera opción se realiza mediante cuatro controles del tipo ComboBox, en los que se especifica la región, ciudad, mes y tipo de día. Dichos controles están programados de forma que la introducción de datos siga un orden coherente, por ejemplo, no permite seleccionar ninguna ciudad si no se ha especificado la región en que se haya. Esto se debe a que la lista de ciudades disponibles varía en función de la región seleccionada. La segunda opción se ejecuta haciendo click sobre un control CheckBox, el cual abre un panel de introducción manual de datos.

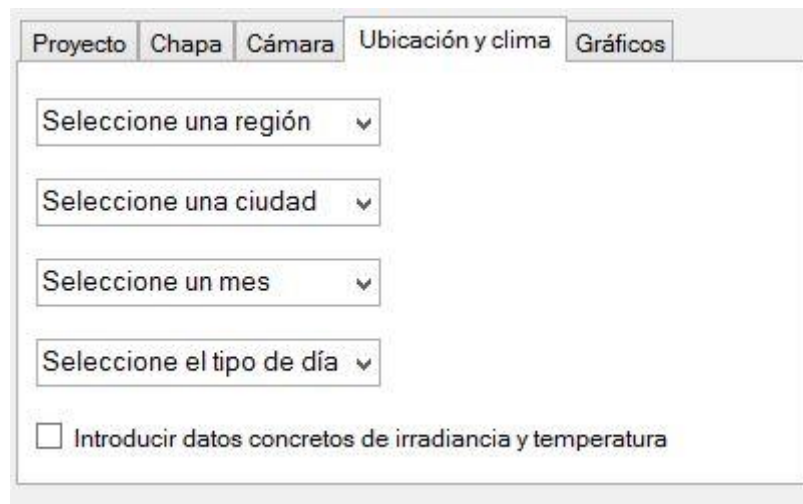


Figura 12: Pestaña de introducción de parámetros ambientales

- 5) **Gráficos:** permite al usuario graficar los resultados de los cálculos, ofreciéndoles distintas combinaciones de variables. Cuenta con dos controles del tipo ComboBox, cuya función es especificar cual de las listas de variables se corresponde con el eje de abscisas y cual con el de ordenadas. Las variables se muestran como listas por medio de controles ListBox. En la lista del eje de ordenadas únicamente se permite la selección de una de las tres variables, mientras que la lista de ordenadas permite realizar una selección múltiple.

La parte derecha de esta pestaña ofrece una serie de controles que permiten al usuario especificar el tipo de gráfico que desean generar. Para ello cuenta con tres controles NumericUpDown, tres controles Button y dos controles CheckBox. Los controles del tipo NumericUpDown permiten establecer los valores máximo y mínimo de la variable asignada al eje de abscisas, así como especificar el intervalo entre dos puntos consecutivos de dicha variable. Cada control de este tipo viene acompañado por un control de tipo Label, en el cual se especifica su función. Los controles CheckBox permiten añadir título y leyenda al gráfico. Por último, los controles del tipo Button permiten especificar el tipo de gráfico. Ofrecen tres opciones: de barras, de tipo interpolatorio o comparativa entre distintos materiales. Esta última opción genera gráficos de tipo interpolatorio en los que se compara, bajo unas mismas condiciones, los cuatro materiales predefinidos disponibles en la pestaña Chapa.

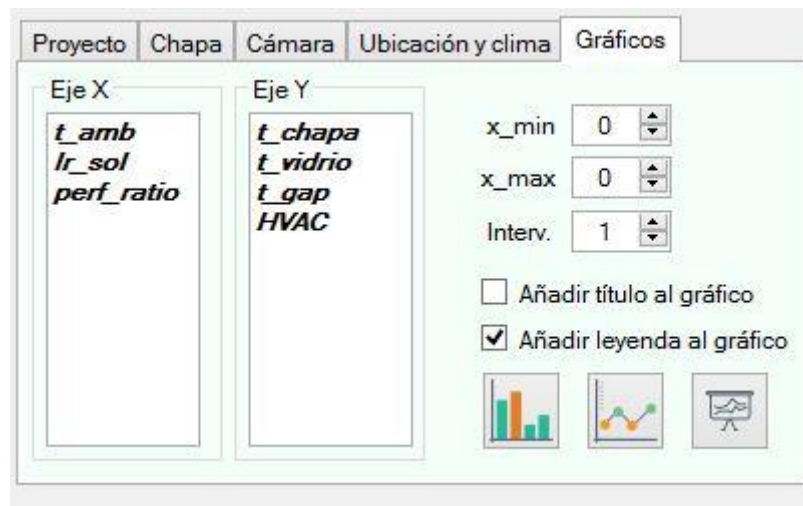


Figura 13: Pestaña de gráficos

grpDatos

Se trata de un control de tipo `GroupBox` cuya función es contener a `tblInputData` y señalar que se trata de la zona de manipulación de datos en el ejecutable.

trackIrradiancia

Se trata de un control de tipo `TrackBar` cuya función es permitir al usuario variar las condiciones de irradiancia solar. La función de este control es redundante, ya que las condiciones de irradiancia solar se pueden concretar en la pestaña de Ubicación y clima del control `tblInputData`. No obstante, el objetivo de este control es permitir una variación rápida y visual de dichas condiciones. Además, viene acompañado por un control tipo `Label` que especifica que se trata de la barra de irradiancia y un control tipo `TextBox`, el cual muestra en todo momento el valor actual de la irradiancia con sus unidades correspondientes.

trackPerfRate

Se trata de un control de tipo `TrackBar` cuya función es idéntica a la de `trackIrradiancia` con la única diferencia de que permite variar el valor del ratio de perforación de la chapa en lugar del nivel de irradiancia solar.

trackTamb

Se trata de un control de tipo TrackBar cuya función es idéntica a la de *trackIrradiancia* y *trackPerfRate* con la única diferencia de que permite variar el valor de la temperatura ambiente.

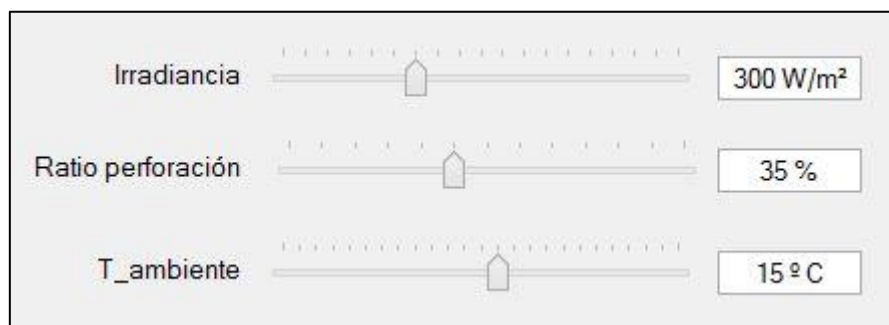


Figura 14: Controles *trackIrradiancia*, *trackPerfRate* y *trackTamb*

checkAutomatico

Se trata de un control de tipo CheckBox cuya función es permitir al usuario especificar si desea que el ejecutable realice los cálculos de forma automática a medida que va introduciendo las variables o si, por el contrario, desea ser él quien decida cuando debe el ejecutable realizar los cálculos.

checkSinChapa

Se trata de un control de tipo CheckBox que permite realizar los cálculos eliminando la chapa, es decir, considerando un edificio con fachada simple de vidrio.

checkChapaMaciza

Se trata de un control de tipo CheckBox que permite realizar los cálculos considerando que la chapa es maciza en lugar de perforada.

btnCalcular

Se trata de un control de tipo Button que ejecuta el algoritmo de cálculo y muestra los resultados en pantalla al hacer click sobre él. El icono de este control es una calculadora.

btnConsumoAhorro

Se trata de un control de tipo Button que, al hacer click sobre el, muestra en pantalla la estimación de consumo en iluminación, calefacción y aire acondicionado del edificio, en kWh. Los valores utilizados para realizar dichas estimaciones han sido obtenidos experimentalmente en dos edificios reales con fachada de doble piel de chapa perforada; uno de ellos ubicado en Bilbao y el otro en Sevilla. Por ello, este botón no es visible si en la pestaña de Ubicación y clima del control *tabInputData* no se ha seleccionado alguna de estas dos ciudades. El icono de este control es el símbolo del euro.

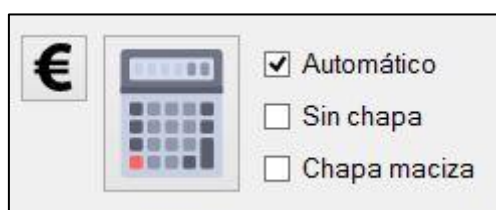


Figura 15: Controles *checkAutomatico*, *checkSinChapa*, *checkChapaMaciza*, *btnCalcular* y *btnConsumoAhorro*

pic30cm

Se trata de un control de tipo PictureBox que muestra el esquema constructivo de la fachada en el caso de que la anchura de la cámara sea de 30cm.

pic5cm

Se trata de un control de tipo PictureBox que muestra el esquema constructivo de la fachada en el caso de que la anchura de la cámara sea de 5cm.

lblTchapa

Se trata de un control de tipo Label cuya función es mostrar en pantalla la temperatura de la chapa calculada por el ejecutable con sus unidades correspondientes. Este control no es visible si *checkSinChapa* está seleccionado.

lblTvidrio

Se trata de un control de tipo Label cuya función es mostrar en pantalla la temperatura del vidrio calculada por el ejecutable con sus unidades correspondientes.

lblTgap

Se trata de un control de tipo Label cuya función es mostrar en pantalla la temperatura de la cámara calculada por el ejecutable con sus unidades correspondientes. Este control no es visible si *checkSinChapa* está seleccionado.

lblHVAC

Se trata de un control de tipo Label cuya función es mostrar en pantalla el consumo de HVAC calculado por el ejecutable con sus unidades correspondientes. Su contenido cambia de color en función de si implica retirar calor (azul) o aportarlo (rojo).

picHVAC

Se trata de un control de tipo PictureBox cuya función es puramente estética. Muestra una imagen que simboliza el calor y el frío rodeando al control *lblHVAC*.

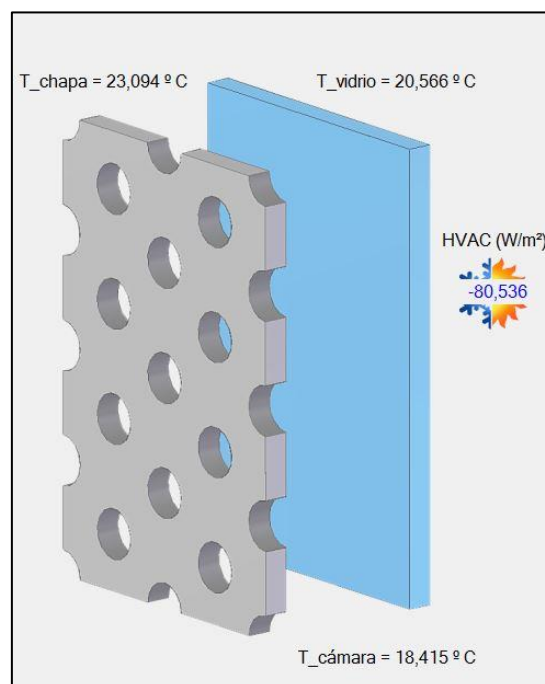


Figura 16: Controles *pic30cm*, *lblTchapa*, *lblTvidrio*, *lblTgap*, *lblHVAC* y *picHVAC*

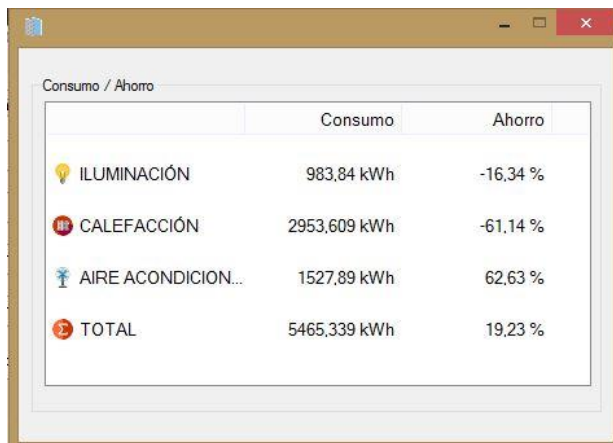
5.3.1.2. FormConsumoAhorro

La función de FormConsumoAhorro es mostrar los resultados de la estimación del consumo de iluminación, calefacción y aire acondicionado que realiza el ejecutable.

Dicha estimación está basada en datos experimentales tomados en dos edificios reales con fachada de doble piel de chapa perforada: uno en Bilbao y el otro en Sevilla. Se accede a FormConsumoAhorro haciendo click sobre *btnConsumoAhorro*.

El motivo por el que estos datos se consultan en un formulario distinto al principal es que, como se ha explicado con anterioridad, los datos en los que se basan las estimaciones son experimentales y, por lo tanto, solo se ajustan a las ubicaciones concretas de Bilbao y Sevilla. Para todo el resto de ubicaciones y condiciones posibles el ejecutable no posee datos a partir de los cuales poder estimar el consumo, por lo que en la gran mayoría de los casos posibles no genera ningún resultado.

El aspecto que presenta FormConsumoAhorro se muestra en la [Figura 17](#).



	Consumo	Ahorro
ILUMINACIÓN	983,84 kWh	-16,34 %
CALEFACCIÓN	2953,609 kWh	-61,14 %
AIRE ACONDICION...	1527,89 kWh	62,63 %
TOTAL	5465,339 kWh	19,23 %

Figura 17: FormConsumoAhorro

Para cumplir su función, FormConsumoAhorro cuenta con los dos siguientes controles:

lviConsumoAhorro

Se trata de un control de tipo ListView que muestra los resultados de las estimaciones del consumo en iluminación, calefacción y aire acondicionado. Para ello, el control cuenta con tres columnas. En la primera de ellas, se especifica a con qué tipo de consumo se corresponde cada una de las filas. La segunda, encabezada con el título de Consumo, muestra el consumo (en kWh). Por último, en la tercera columna se muestra el ahorro que supone la fachada de doble piel de chapa perforada frente a un sistema de fachada simple.

grpConsumoAhorro

Se trata de un control de tipo GroupBox cuya función es contener a *lviConsumoAhorro*.

5.3.1.3. FormIrradTamb

La función de FormIrradTamb es servir como panel que permita al usuario la introducción manual de datos climatológicos. Los datos que admite son el nivel de irradiancia solar y la temperatura ambiente. Se accede a FormIrradTamb seleccionando la opción “Introducir datos concretos de irradiancia y temperatura” en la página Ubicación y clima de *tabInputData*.

El aspecto que presenta FormIrradTamb se muestra en la [Figura 18](#).



Figura 18: FormIrradTamb

Los controles que utiliza FormIrradTamb para cumplir su función son los siguientes:

numIrradiancia

Se trata de un control de tipo NumericUpDown cuya función es asignar el valor de irradiancia solar que posteriormente será utilizado en los cálculos. Este control permite una precisión de dos decimales y va acompañado por dos controles Label que indican el tipo de dato que el usuario está introduciendo y sus unidades.

numTamb

Se trata de un control de tipo NumericUpDown cuya función es idéntica a la de *numIrradiancia* con la única diferencia de que el parámetro que permite especificar es la temperatura ambiente.

btnAceptar

Se trata de un control de tipo Button cuya función es validar los datos introducidos y cerrar el formulario.

btnCancelar

Se trata de un control de tipo Button cuya función es cerrar el formulario sin validar los datos introducidos. En caso de pulsar este botón, los valores que estuviesen asignados antes de acceder a FormIrradTamb se mantendrán intactos.

5.3.1.4. FormUserDefMaterial

La función de FormUserDefMaterial es servir como panel que permita al usuario la introducción manual de parámetros del material de la chapa. En este sentido, su función y estructura son muy similares a las de FormIrradTamb. Los parámetros que permite que el usuario introduzca son la absorptividad y emisividad del material de la chapa. FormUserDefMaterial es accesible haciendo click sobre la opción “*User defined material*” de la página Chapa de *tblInputData*.

El aspecto que presenta FormUserDefMaterial se muestra en la [Figura 19](#).

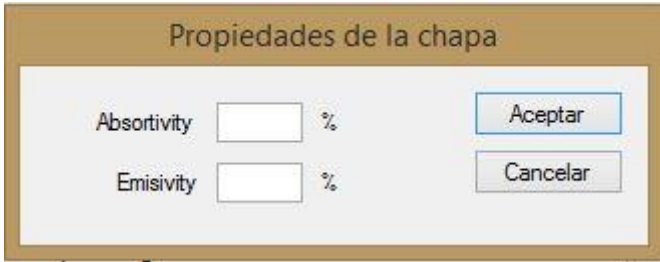


Figura 19: FormUserDefMaterial

Los controles que utiliza FormUserDefMaterial para cumplir su función son los siguientes:

txtAlpha

Se trata de un control de tipo TextBox que permite al usuario especificar la absorptividad del material de la chapa. Este control va acompañado por dos controles de tipo Label, los cuales especifican el parámetro que permite introducir y que éste debe ser introducido como un porcentaje.

txtEpsilon

Se trata de un control de tipo TextBox cuya función es idéntica a la de *txtAlpha* con la única diferencia de que, en este caso, el parámetro que permite introducir es la emisividad del material.

btnAceptar

Se trata de un control de tipo Button cuya función es validar los datos introducidos y cerrar el formulario.

btnCancelar

Se trata de un control de tipo Button cuya función es cerrar el formulario sin validar los datos introducidos. En caso de pulsar este botón, los valores que estuviesen asignados antes de acceder a FormIrradTamb se mantendrán intactos.

5.3.1.5. FormGraficos

La función de FormGraficos es mostrar los graficados los datos que el usuario especifique. Además, permite también guardar los gráficos mediante un control Button en la zona inferior izquierda. Dicho botón se reconoce por el icono del disquete, habitualmente utilizado en las aplicaciones para las opciones que permiten guardar. Su interfaz puede presentar distintos aspectos, en función del tipo de gráfico que el usuario haya generado. Por un lado, cuando el gráfico es un gráfico sencillo de barras o de tipo interpolatorio, el formulario muestra dicho gráfico en la zona central y, si así se ha especificado, un título en la parte superior y una leyenda a la derecha. Por otro lado, en el caso de que se genere un gráfico de comparativa de materiales, pueden darse dos situaciones:

- La variable del eje de abscisas es el ratio de perforación: en este caso, FormGraficos posee, además del área del gráfico, cuatro controles de tipo Button y otros cuatro de tipo ComboBox.

Los cuatro controles Button permiten seleccionar cual de las variables se muestra en el eje de ordenadas. Además, al hacer click sobre cualquiera de los cuatro botones tan solo existe un área de gráficos.

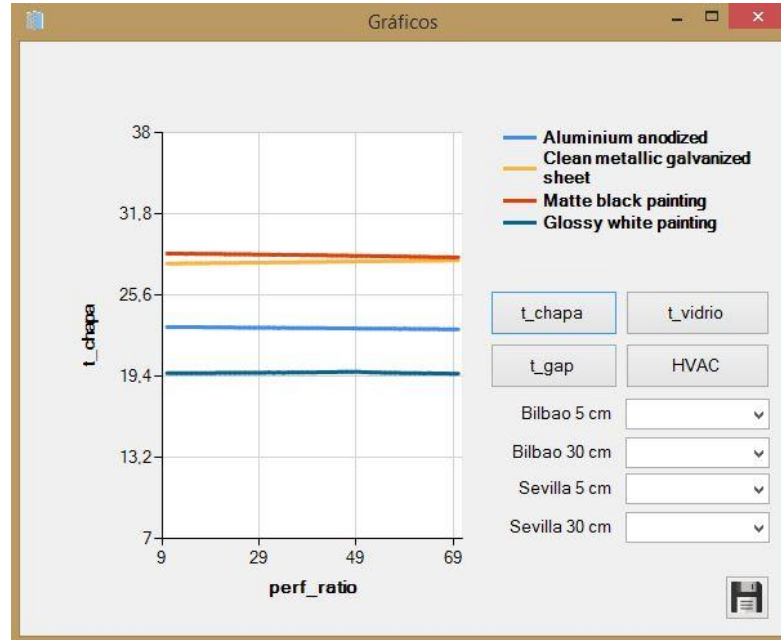


Figura 20: Gráfico de comparación simple

Los cuatro controles ComboBox permiten seleccionar entre las ubicaciones de Bilbao o Sevilla, y anchura de la cámara de 5 o 30 centímetros. Además, al desplegarlas permite, sobre la opción seleccionada, escoger entre consumo en iluminación, en calefacción o en aire acondicionado. Al hacerlo, el área del gráfico se divide en dos. El gráfico superior muestra el consumo en función del ratio de perforación, y el inferior muestra el ahorro.

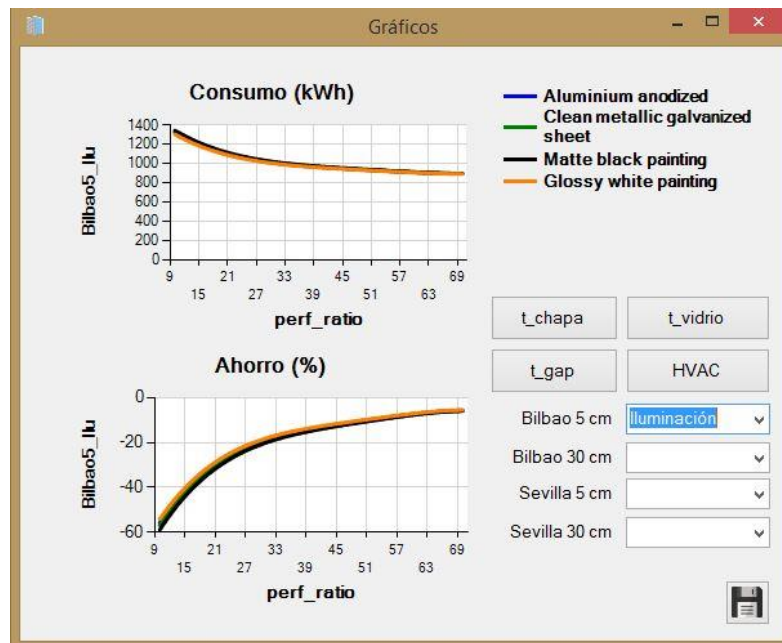


Figura 21: Gráfico de comparación doble

- La variable en el eje de abscisas NO es el ratio de perforación: en este caso, FormGraficos muestra solo el área del gráfico y los cuatro controles Button del caso anterior, esta vez sin los controles ComboBox. Por lo demás, el aspecto que presenta es idéntico al de la Figura 20.

Los controles que permiten mostrar los gráficos son de tipo Chart, y FormGraficos posee los dos siguientes:

chartGrafico

Este control posee una sola área para graficar y es el que se utiliza para mostrar los gráficos simples, tanto los de barras como los de tipo interpolatorio.

chartCompararMateriales

Este control funciona con una o dos áreas de gráficos, en función del tipo de comparación que el usuario realice. Es el que se muestra al seleccionar la opción de comparar materiales en la pestaña de Gráficos de *tabInputData*.

5.3.1.6. MessageBox e InputBox

Las funciones MessageBox e InputBox no forman parte de la interfaz gráfica directamente, ya que no son controles ni formularios. Se trata de construcciones

gráficas predefinidas en VB.NET que se generan desde el código, pero que actúan de forma similar a un formulario.

La función MessageBox genera una ventana emergente que contiene un mensaje para el usuario. Dicho mensaje puede ser de advertencia, error, información o una pregunta. Además, admite opciones de validación o cancelación. Esta función es utilizada en el ejecutable para generar todos los mensajes de advertencia, error e información que se dan en tiempo de ejecución. En la [Figura 22](#) se muestra un ejemplo de MessageBox.

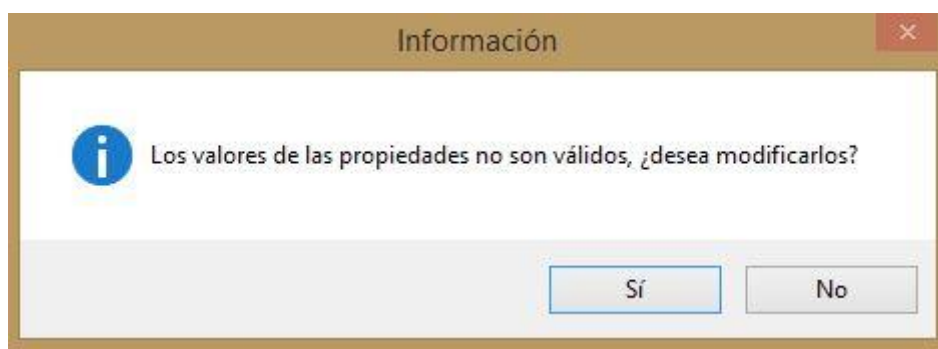


Figura 22: Ejemplo de MessageBox

La función InputBox, al igual que MessageBox, genera una ventana emergente. No obstante, el cometido de esta ventana no es generar un mensaje sino permitir al usuario la introducción de un valor concreto (ver [Figura 23](#)).

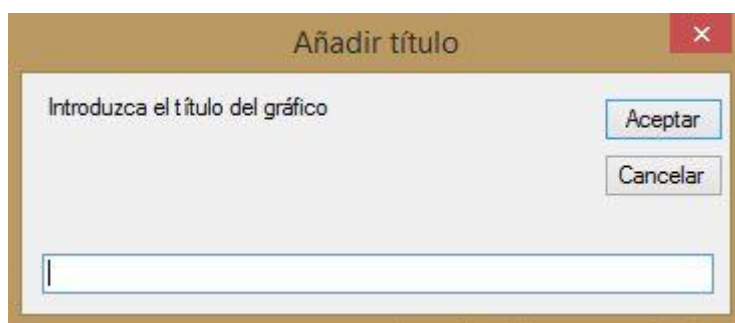


Figura 23: Ejemplo de InputBox

5.3.2. Código escrito

En este subapartado se analiza el código que hace que el ejecutable y todos los controles de la interfaz funcionen. Se comienza con una explicación de la estructura de código de VB.NET, y de sus diferencias con respecto a la programación con lenguajes

tradicionales. A continuación se explican los conceptos de módulo y clase, que son los dos tipos de fichero de código con los que trabaja la aplicación. Finalmente, se realiza un análisis exhaustivo de los módulos y clases que componen el código, analizando su finalidad, las funciones y procedimientos que contienen y lo que cada una de estas hace.

Con los lenguajes de programación tradicionales, también conocidos como lenguajes de procedimiento, el programa en sí dicta por completo qué código se ejecuta en cada momento. Cuando se ejecuta un programa de este tipo, se ejecuta la primera línea de código del programa y continúa su ejecución siguiendo una ruta totalmente predeterminada. La ejecución del código se puede ramificar o entrar en un bucle ocasionalmente, pero la ruta viene totalmente determinada por el programa. Esto difiere del comportamiento de una aplicación de Windows, en la que el usuario puede interactuar con cualquier parte de la interfaz en el orden que escoja [9].

VB.NET, por su parte, incorpora un módulo de programación orientada a eventos. Los eventos pueden definirse como los cambios de estado producidos en los objetos de la interfaz, que son debidos a la interacción del usuario con la misma. Un ejemplo claro es el cambio de un control CheckBox que, al hacer click sobre él, pasa de estar en blanco a estar seleccionado.

Como el ejecutable desarrollado es una aplicación orientada a eventos, no se ve limitado por las restricciones de los programas de lenguajes de procedimiento. El código del ejecutable está compuesto por secciones lógicas asociadas a eventos, que se ejecutan al activarse el evento al que están asociadas. En este sentido, el usuario posee un control total sobre el orden en que se ejecutan las sentencias del código, activando de forma interactiva eventos específicos tales como hacer click sobre un botón.

El código del ejecutable está contenido en módulos, los cuales son ficheros de texto en los que se almacena el código al escribirlo. Existen dos tipos de módulos: los módulos estándar (en adelante, simplemente módulos) y los módulos de clase (en adelante, clases). Los módulos se utilizan para agrupar conjuntos relacionados de código, de forma que éste esté más ordenado y sea más fácil de comprender. Las clases, por su parte, se utilizan como plantillas para la instalación de objetos [9].

Como se puede apreciar en la [Figura 24](#), el código del ejecutable está agrupado en tres módulos (modCalculos, modGraficos y modDatos) y en cinco clases, que se corresponden con cada uno de los cinco formularios analizados en la sección anterior. Dentro de los módulos, el código está agrupado en procedimientos y funciones. Ambos son rutinas de código aisladas que se pueden ejecutar [9], con la diferencia de que una función es una agrupación de código que devuelve un valor concreto que se asigna a una variable, mientras que un procedimiento se ejecuta sin necesidad de devolver ningún valor o valores concretos.

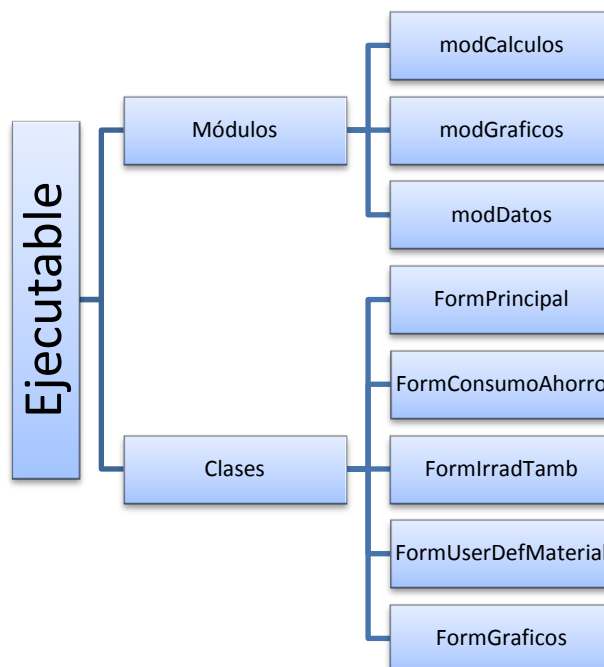


Figura 24: Esquema del código

El código concreto que contiene cada uno de los módulos y clases se encuentra en el [Anexo I](#), el cual ha sido directamente obtenido del IDE en el que se ha compilado la aplicación.

A continuación se realiza un análisis de cada uno de los tres módulos. Se explica su razón de ser, y el funcionamiento de cada una de las funciones y procedimientos que contienen. En rasgos generales, el código contenido en estos tres módulos es el motor de la aplicación, ya que en ellos están todas las funciones y procedimientos de cálculo, gráficos y manejo de datos que dan utilidad al ejecutable. El código escrito en las clases establece la conexión entre los eventos y las funciones y procedimientos, de forma que éstas se ejecuten cuando deben y con los parámetros adecuados.

5.3.2.1. modCalculos

El módulo modCalculos está concebido para contener a aquellas funciones y procedimientos de la aplicación que realicen cálculos numéricos tales como resolver sistemas de ecuaciones, calcular el valor de ciertos parámetros, etc. En ocasiones, como se verá a la hora de analizar dichas funciones y procedimientos por separado, contienen también sentencias de código que alteran alguno de los controles gráficos de los formularios. Esto ofrece la ventaja de que no solo realizan los cálculos pertinentes, sino que además muestran los resultados donde corresponde.

Las funciones que contiene modCalculos son:

$$x = \text{linsolve_LU}(A, b)^8$$

La función *linsolve_LU* resuelve un sistema lineal de ecuaciones de la forma $A \cdot x = b$ por medio del algoritmo de factorización LU⁹. Como parámetros de entrada coge la matriz de coeficientes del sistema (A) y el vector de coeficientes libres (b). La matriz A es una matriz de 2×2 y el vector b es un vector columna de 2×1 .

Esta función complementa a *Calcular_tchapa_vidrio*, la cual necesita resolver un sistema de estas características en una de sus sentencias.

$$value = \text{Signo}(x)$$

La función *Signo* devuelve un valor $value = -1$ en caso de que x sea menor que cero y $value = 1$ si x es mayor que cero. Esta operación se da en numerosas ocasiones en el código, por lo que se opta por implementarla como función con el fin aportar claridad al código y ahorrar líneas.

$$x = \text{Calcular_tchapa_vidrio}(x0, \text{epsilon_IR_chapa}, \text{epsilon_IR_vidrio}, p, t_{amb}, t_{int}, \text{alpha_solar_chapa}, \text{alpha_solar_vidrio}, I_r \text{ sol})$$

La función *Calcular_tchapa_vidrio* aplica el algoritmo de Newton-Raphson¹⁰ para resolver el sistema de ecuaciones resultante de aplicar las funciones de Matlab[®] desarrolladas por *Jesús Mari et al.* para caracterizar la fachada de doble piel como se describe en [4].

El resultado del sistema es el vector x de dimensiones 3×1 que contiene, en este orden, los valores medios de temperatura de la chapa, del vidrio y de la cámara. El significado de los parámetros de entrada se resume en la [Tabla 3](#):

PARÁMETRO	DETALLES
x0	Vector de aproximación inicial. Dimensiones 3×1 .
epsilon_IR_chapa	Emisividad de la chapa. Es función del material y el color.
epsilon_IR_vidrio	Emisividad del vidrio.
p	Ratio de perforación de la chapa expresado en tanto por uno.

⁸ La sintaxis utilizada es descriptiva y no se corresponde con la sintaxis de declaración de funciones o procedimientos en VB.NET. No obstante, los nombres de las funciones y variables entre paréntesis sí que son iguales que en el código.

⁹ Ver 5.4.1.

¹⁰ Ver 5.4.1.

t_amb	Temperatura ambiente en grados Kelvin.
t_int	Temperatura interior del edificio en grados Kelvin.
alpha_solar_chapa	Absortividad de la chapa. Es función del material y del color.
alpha_solar_vidrio	Absortividad del vidrio.
Ir_sol	Irradiancia solar en W/m ²

Tabla 3: Parámetros de entrada de la función *Calcular_tchapa_tvidrio*

Todos los parámetros que aparecen en la [Tabla 3](#) son numéricos o vectores de valores numéricos.

carga_y_ahorro = Calcular_Ilu_Heat_AC (ciudad, material, p, separacion)

La función *Calcular_Ilu_Heat_AC* estima los valores de consumo y ahorro de iluminación, calefacción y aire acondicionado. El vector *carga_y_ahorro* es un vector unidimensional de seis posiciones, cada una de las cuales almacena una variable de tipo String (cadena de caracteres). Las tres primeras posiciones del vector se corresponden con los datos de consumo (en kWh) y las tres últimas con los datos de ahorro (en %). El orden que siguen es: iluminación, calefacción y aire acondicionado. Los parámetros de entrada se resumen en la [Tabla 4](#):

PARÁMETRO	DETALLES
ciudad	Ciudad en la que se encuentra el edificio. Admite Bilbao o Sevilla.
material	Material y color de la chapa.
p	Ratio de perforación de la chapa en tanto por uno.
separacion	Anchura de la cámara. Admite 5cm o 30cm.

Tabla 4: Parámetros de entrada de la función *Calcular_Ilu_Heat_AC*

irrad_tamb = Determinar_Ir_Tamb_Ciudad (ciudad, mes)

La función *Determinar_Ir_Tamb_Ciudad* devuelve los valores de irradiancia solar y temperatura ambiente medios de la ciudad seleccionada por el usuario en el mes especificado. El vector *irrad_tamb* es un vector unidimensional de dos posiciones. La primera de ellas se corresponde con el valor de la irradiancia solar en W/m² y la segunda con la temperatura ambiente en grados Kelvin.

Por otro lado, el módulo modCalculos contiene un único procedimiento, que es:

Hacer_calculos ()

El procedimiento *Hacer_calculos* está pensado para facilitar la llamada al resto de funciones de cálculo del programa. No requiere ningún parámetro de entrada y lo único que hace es actuar como atajo para evitar escribir líneas de código innecesarias en las clases. Cada vez que un evento requiere que se ejecute alguna de las funciones de cálculo, llama al procedimiento *Hacer_calculos* y éste ejecuta las funciones correspondientes. Además, este procedimiento también actualiza los valores que se muestran en *IviConsumoAhorro* de forma automática.

5.3.2.2. modGraficos

El módulo modGraficos está concebido para contener las funciones y procedimientos necesarios generar y organizar los datos para poder graficarlos. Posee también un procedimiento que posiciona *trackIrradiancia*, *trackPerfRate* y *trackTamb* de forma que se hallen en todo momento mostrando el valor de la irradiancia, el ratio de perforación y la temperatura ambiente actualizado.

Las funciones que contiene modGraficos son:

HVAC = Calcular HVAC para graficar (t chapa, t vidrio)

La función *Calcular_HVAC_para_graficar* calcula el calor que debe aportar o extraer el sistema de ACS del edificio en función del balance de calor en la fachada. Para ello, recibe como parámetros de entrada las temperaturas de la chapa y del vidrio, y devuelve HVAC en W/m². Esta función es utilizada por *Generar_grafico* en cada una de sus iteraciones para calcular el valor de HVAC en cada punto que vaya a ser graficado.

boEstaIncluido = Incluye_nombre (nombre, array)

La función *Incluye_nombre* devuelve el valor *boEstaIncluido = True* si *nombre* está forma parte de la lista de nombres *array*. Si por el contrario *nombre* no está incluido en *array*, devuelve el valor *boEstaIncluido = False*. Esta función forma parte de la lógica que sigue el procedimiento *Generar_grafico* una vez que el usuario ha especificado las variables que desea plotear. La variable de salida, *boEstaIncluido*, es una variable de tipo booleano, mientras que *nombre* es de tipo String y *array* es de tipo lista de String.

comp_mat = Valores_grafico_materiales (vble x, xmin, xmax)

La función *Valores_grafico_materiales* recibe como entrada el parámetro que el usuario haya especificado como variable del eje de abscisas junto con sus valores

máximo y mínimo. Con ellos, genera una matriz *comp_mat*, la cual contiene los valores de temperatura de la chapa, temperatura del vidrio, temperatura de la cámara y HVAC que se corresponden para cada punto del array *vble_x* comprendido entre *xmin* y *xmax*. Esta matriz es utilizada por *Grafico_materiales* para generar el gráfico de comparativa de los distintos materiales en *FormGraficos*.

cargas_y_ahorro = Cargas_Ilu_Heat_AC_para_graficar (xmax, xmin)

La función *Cargas_Ilu_Heat_AC_para_graficar* tiene un cometido similar al de *Valores_grafico_materiales*, solo que en lugar de devolver una matriz con los valores de temperaturas y HVAC, devuelve la matriz *cargas_y_ahorro* que contiene las estimaciones de consumo y ahorro en iluminación, calefacción y aire acondicionado. Esta función es utilizada por *Grafico_materiales* a la hora de generar los gráficos que comparan el consumo y el ahorro de los distintos materiales.

Por otro lado, los procedimientos que contiene *modGraficos* son:

Posicionar_barras (lr_sol, p, t_amb)

El procedimiento *Posicionar_barras* se encarga de que *trackIrradiancia*, *trackPerfRate* y *trackTamb* estén en todo momento en la posición que les corresponde, y mostrando el valor actualizado de la irradiancia, el ratio de perforación y la temperatura ambiente respectivamente.

Generar_grafico (vble_x, vbles_y, xmin, xmax, tipo_grafico, titulo_grafico)

El procedimiento *Generar_grafico* es el encargado de generar los gráficos simples una vez que el usuario haga click sobre el botón de gráfico de barras o de gráfico de tipo interpolatorio. Con anterioridad en este mismo apartado se han expuesto y comentado las funciones que utiliza este procedimiento para obtener todos los valores que requiere. Los parámetros que *Generar_grafico* requiere de entrada están resumidos en la [Tabla 5](#).

PARÁMETRO	DETALLES
<i>vble_x</i>	Indica la variable que representa el eje de abscisas.
<i>vbles_y</i>	Indica la(s) variable(s) que representa el eje de ordenadas.
<i>xmin</i>	Indica el valor mínimo del eje de abscisas.
<i>xmax</i>	Indica el valor máximo del eje de abscisas.
<i>tipo_grafico</i>	Indica el tipo de gráfico: de barras o de tipo interpolatorio.
<i>titulo_grafico</i>	Indica el título del gráfico.

Tabla 5: Parámetros de entrada del procedimiento *Generar_grafico*

Grafico_materiales (vble_x, vble_y, titulo_grafico, xmax, xmin)

El procedimiento *Grafico_materiales* tiene el mismo cometido que *Generar_grafico* con la diferencia de que éste genera el gráfico de comparativas entre distintos materiales. Los parámetros de entrada son similares, solo que en este caso solo acepta una variable en el eje de ordenadas. Esto se debe a que cada variable se grafica cuatro veces, una por cada material escogido.

5.3.2.3. modDatos

El módulo *modDatos* está concebido para contener las funciones y procedimientos que permiten al ejecutable generar proyectos, manipular su contenido y guardar ficheros con los datos de cada proyecto. En este caso, el módulo *modDatos* cuenta solo con procedimientos, que son los que se describen a continuación.

Asignar_datos_al_proyecto (nombre)

El procedimiento *Asignar_datos_al_proyecto* traduce la información contenida en el resumen de un proyecto para convertirla en variables internas del ejecutable. El parámetro de entrada *nombre* hace referencia al nombre del proyecto del cual se extrae la información. Conociendo el nombre del proyecto, el procedimiento convierte la información contenida en el resumen desplegable de dicho proyecto en variables, y fuerza al ejecutable a recalcular las temperaturas.

Rellenar_proyecto (nombre)

El procedimiento *Rellenar_proyecto* realiza la tarea inversa que *Asignar_datos_al_proyecto*. Este procedimiento traduce las variables del ejecutable para convertirlas en información que se muestra en pantalla en el resumen desplegable del proyecto *nombre*.

Generar_proyecto (nombre)

El procedimiento *Generar_proyecto* crea un proyecto nuevo de nombre "*nombre*". Al hacerlo, genera un nuevo proyecto en el resumen y en él carga los datos de ejecución del programa en el momento de la creación del proyecto.

Guardar_proyecto ()

Este procedimiento guarda la información del proyecto seleccionado en un fichero de datos. La extensión del fichero generado es propia de este ejecutable (.aadm) y la información en él está estructurada de forma tal que pueda ser reconocible para el ejecutable al pulsar el botón de abrir.

Abrir proyecto ()

Este procedimiento permite al ejecutable leer la información procedente de un fichero .aadm y cargarla como proyecto. Al ejecutarse, genera un nuevo proyecto y, además, fuerza al ejecutable a recalcular todos los resultados, de forma que se correspondan con los datos de entrada del nuevo proyecto.

5.4. Algoritmos matemáticos

En este subapartado se da una explicación detallada de los algoritmos en los que se basan las funciones de cálculo del ejecutable. En primer lugar, se expone el algoritmo de resolución de sistemas de ecuaciones que utiliza la función *Calcular_tchapa_vidrio* para obtener las temperaturas de la chapa y del vidrio, así como su origen. A continuación se muestra el método de cálculo empleado para obtener los valores de demanda de HVAC del edificio. Por último, se expone el método seguido para realizar la estimación del consumo de iluminación, calefacción y aire acondicionado.

5.4.1. Cálculo de las temperaturas de la chapa y el vidrio

El algoritmo matemático implementado en la función *Calcular_tchapa_vidrio* aplica el método de Newton-Raphson para la resolución del sistema de ecuaciones no lineales descrito en el apartado 5.2.

Originalmente, dicho sistema fue implementado en dos funciones escritas en Matlab® por *Jesús Mari et al.* para proceder a su resolución. Como el ejecutable desarrollado está escrito en lenguaje VB.NET, se hace necesaria la transcripción e implementación del código de dichas funciones de Matlab® a código VB.NET. Esta transcripción presenta un problema: el algoritmo de resolución del sistema de ecuaciones. Matlab® cuenta con funciones y métodos propios para abordar la resolución de sistemas de estas características, cosa que no sucede en VB.NET. Por lo tanto, se opta por implementar el método de Newton-Raphson en el ejecutable proceder a la resolución del sistema y obtener así las temperaturas de la chapa y el vidrio. El método de Newton-Raphson es el que se describe a continuación:

Método de Newton-Raphson¹¹

¹¹ Método tomado de [12]

De forma general, se puede expresar el problema de resolución de un sistema de N ecuaciones no lineales con N incógnitas como:

$$F_i(x_0, x_1, \dots, x_{N-1}) = 0 \quad i = 0, 1, \dots, N - 1$$

Denotando como \mathbf{x} al vector de valores x_i y como \mathbf{F} al vector de funciones F_i y realizando la expansión en serie de Taylor de cada una de las funciones F_i se obtiene:

$$F_i(\mathbf{x} + \delta\mathbf{x}) = F_i(\mathbf{x}) + \sum_{j=0}^{N-1} \frac{\partial F_i}{\partial x_j} \delta x_j + O(\delta\mathbf{x}^2)$$

La matriz de derivadas parciales que se obtiene en la ecuación anterior es la matriz Jacobiana \mathbf{J} :

$$J_{ij} = \frac{\partial F_i}{\partial x_j}$$

Denotando la ecuación de la expansión en serie de Taylor en notación matricial:

$$\mathbf{F}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{F}(\mathbf{x}) + \mathbf{J} \cdot \delta\mathbf{x} + O(\delta\mathbf{x}^2)$$

Despreciando los términos de orden $\delta\mathbf{x}^2$ y superiores e igualando $\mathbf{F}(\mathbf{x} + \delta\mathbf{x}) = 0$, se obtiene un sistema de ecuaciones lineales para obtener las correcciones $\delta\mathbf{x}$ que simultáneamente aproximan cada una de las funciones F_i a cero:

$$\mathbf{J} \cdot \delta\mathbf{x} = -\mathbf{F}$$

Dicho sistema lineal se puede resolver de múltiples maneras, como por ejemplo por factorización LU, de forma que al vector solución \mathbf{x} se le añade la corrección:

$$\mathbf{x}_{nuevo} = \mathbf{x}_{viejo} + \delta\mathbf{x}$$

Este último proceso se itera hasta alcanzar la convergencia del sistema.

Se ha escogido el método de Newton-Raphson entre las múltiples opciones existentes por su sencillez y velocidad de convergencia. No obstante, este método presenta dos desventajas:

- Su convergencia no es absoluta, por lo que la aproximación inicial debe ser cercana a la solución final.
- Presenta la dificultad añadida de requerir el cálculo de la matriz Jacobiana del sistema.

Para proceder al cálculo de la matriz Jacobiana, se contemplan dos opciones: el cálculo analítico o el cálculo numérico. La determinación analítica de la matriz Jacobiana es laboriosa y ha de ser realizada antes de ser implementada en VB.NET. No obstante, tiene la ventaja de ser absolutamente precisa y, una vez determinada, requiere un mínimo coste computacional. Los métodos de derivación numérica, por su parte, presentan la desventaja de una mayor complejidad en su implementación y de posibles problemas de precisión. No obstante son soluciones mucho más versátiles, ya que soportan cualquier modificación en el sistema. Además, pueden ser implementadas directamente en VB.NET.

Como se cuenta con recursos computacionales suficientes, se opta por determinar la matriz Jacobiana de forma analítica. Para ello, se introduce el sistema de ecuaciones descrito en [4] en el software Wolfram Mathematica®, posee un potente entorno de cálculo simbólico. Una vez obtenida la matriz Jacobiana del sistema, la implementación del método de Newton-Raphson es sencilla y directa.

Para la resolución del sistema lineal resultante de aplicar el método de Newton-Raphson, el ejecutable aplica el método de factorización LU, que se describe a continuación.

Método de factorización LU¹²

En general, se puede denotar un sistema de N ecuaciones lineales en forma matricial de la siguiente manera:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$$

Donde \mathbf{A} es la matriz de coeficientes del sistema y \mathbf{b} el vector de coeficientes libres del sistema. Suponiendo que la matriz \mathbf{A} se puede escribir como el producto de dos matrices:

$$\mathbf{L} \cdot \mathbf{U} = \mathbf{A}$$

Donde \mathbf{L} es una matriz triangular inferior y \mathbf{U} es una matriz triangular superior. En el caso de que \mathbf{A} fuese, por ejemplo, una matriz de 3×3 , podría escribirse como

$$\begin{bmatrix} \alpha_{00} & 0 & 0 \\ \alpha_{10} & \alpha_{11} & 0 \\ \alpha_{20} & \alpha_{21} & \alpha_{22} \end{bmatrix} \cdot \begin{bmatrix} \beta_{00} & \beta_{01} & \beta_{02} \\ 0 & \beta_{11} & \beta_{12} \\ 0 & 0 & \beta_{22} \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix}$$

El modo de abordar la resolución del sistema mediante factorización LU es descomponer el sistema original en dos subsistemas:

$$\mathbf{A} \cdot \mathbf{x} = (\mathbf{L} \cdot \mathbf{U}) \cdot \mathbf{x} = \mathbf{L} \cdot (\mathbf{U} \cdot \mathbf{x}) = \mathbf{b}$$

¹² Método tomado de [12]

De forma que primero se calcula un vector \mathbf{y} tal que

$$\mathbf{L} \cdot \mathbf{y} = \mathbf{b}$$

y después se obtiene \mathbf{x} resolviendo

$$\mathbf{U} \cdot \mathbf{x} = \mathbf{y}$$

La ventaja de descomponer un sistema lineal en dos sistemas de matriz triangular reside en la sencillez de resolución que presentan estos últimos sistemas. El vector \mathbf{y} se puede calcular mediante las expresiones

$$y_0 = \frac{b_0}{\alpha_{00}}$$

$$y_i = \frac{1}{\alpha_{ii}} \left[b_i - \sum_{j=0}^{i-1} \alpha_{ij} y_j \right] \quad i = 1, 2, \dots, N - 1$$

Una vez obtenido \mathbf{y} , el cálculo de \mathbf{x} es

$$x_{N-1} = \frac{y_{N-1}}{\beta_{(N-1),(N-1)}}$$

$$x_i = \frac{1}{\beta_{ii}} \left[y_i - \sum_{j=i+1}^{N-1} \beta_{ij} x_j \right] \quad i = N - 2, N - 3, \dots, 0$$

La función *Calcular_tchapa_vidrio* coge los parámetros de entrada mencionados en [5.3.2.1](#) y, utilizando x_0 como vector de aproximación inicial, aplica el método de Newton-Raphson para obtener las temperaturas de la chapa y el vidrio.

5.4.2. Cálculo de la demanda de HVAC

Una vez obtenidas las temperaturas de la chapa y del vidrio, se dispone de los datos suficientes como para calcular Q_{HVAC} , que simboliza el calor que es necesario extraer o aportar al interior para que este mantenga la temperatura especificada. En sus cálculos, el ejecutable considera una temperatura interna de confort (T_{int}) de 21°C. Para obtener Q_{HVAC} se aplica el balance de calor de la [Figura 25](#).

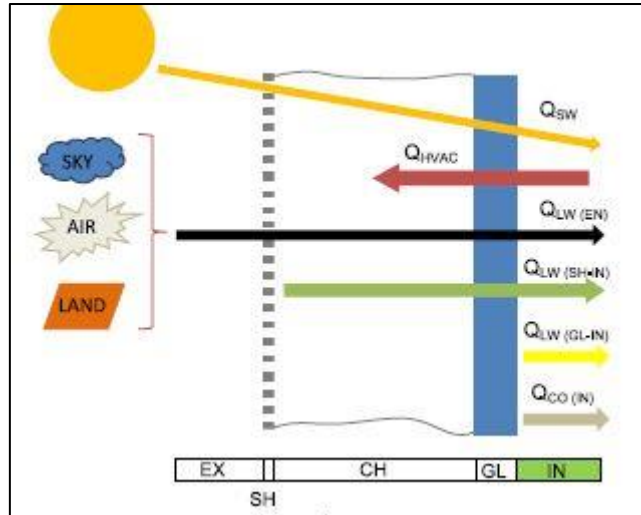


Figura 25: Intercambio de calor entre la fachada y el interior del edificio. (Imagen tomada de [4]).

Para calcular el valor de cada uno de los flujos de calor, el ejecutable los divide en tres grupos: el calor transmitido por convección del vidrio al interior Q_{CO} , el calor de onda corta que llega desde el sol en forma de radiación Q_{SW} y el calor de onda larga intercambiado entre el interior del edificio y el entorno Q_{LW} .

El término Q_{CO} se determina mediante la expresión

$$Q_{CO} = h_{int} \cdot (T_{vidrio} - T_{int})$$

Donde el coeficiente de convección h_{int} es el recomendado para cámaras de aire [13]

$$h_{int} = 1.31 \cdot |T_{vidrio} - T_{int}|^{1/3}$$

El término Q_{SW} se determina por medio de la expresión

$$Q_{SW} = I_{r_{SOL}} \cdot p \cdot \tau_{SW_{vidrio}}$$

Donde $I_{r_{SOL}}$ es el nivel de irradiancia solar, p es el ratio de perforación de la chapa y $\tau_{SW_{vidrio}}$ es la transmisividad de onda corta del vidrio.

El término Q_{LW} se obtiene sumando las contribuciones de calor intercambiado por el interior con el suelo ($Q_{LW_{gnd}}$), con el cielo ($Q_{LW_{sky}}$), con el aire ($Q_{LW_{air}}$), con el vidrio ($Q_{LW_{vidrio}}$) y con la chapa ($Q_{LW_{chapa}}$). De modo que

$$Q_{LW} = (Q_{LW_{air}} + Q_{LW_{sky}} + Q_{LW_{gnd}}) \cdot p \cdot \tau_{LW_{vidrio}} + Q_{LW_{vidrio}} + Q_{LW_{chapa}}$$

Donde p es el ratio de perforación y $\tau_{LWvidrio}$ es la transmisividad de onda larga del vidrio.

Los calores intercambiados con el aire, el cielo y el suelo se obtienen mediante las expresiones

$$Q_{LW_{air}} = \varepsilon_{interior} \cdot \sigma \cdot F_{air} \cdot (T_{air}^4 - T_{int}^4)$$

$$Q_{LW_{sky}} = \varepsilon_{interior} \cdot \sigma \cdot F_{sky} \cdot (T_{sky}^4 - T_{int}^4)$$

$$Q_{LW_{gnd}} = \varepsilon_{interior} \cdot \sigma \cdot F_{gnd} \cdot (T_{gnd}^4 - T_{int}^4)$$

Donde $\varepsilon_{interior}$ es la emisividad del interior, σ es la constante de Boltzmann y los términos F son los factores de forma. El valor de estos últimos se muestra en la [Tabla 6](#).

F_{air}	$0.5 \cdot (1 - 0.5^{1/2})$
F_{sky}	$0.5 \cdot 0.5^{1/2}$
F_{gnd}	0.5

Tabla 6: Valor de los factores de forma del aire, el cielo y el suelo

El calor intercambiado con la chapa se obtiene mediante la expresión

$$Q_{LW_{vidrio}} = \varepsilon_{interior} \cdot \sigma \cdot (T_{vidrio}^4 - T_{int}^4)$$

Por último, el calor intercambiado por la chapa viene dado por la expresión

$$Q_{LW_{chapa}} = \varepsilon_{interior} \cdot \sigma \cdot (T_{chapa}^4 - T_{int}^4) \cdot \tau_{LW_{vidrio}} \cdot (1 - p)$$

Todos los parámetros que definen los materiales de la chapa y el vidrio (emisividad, absorptividad, transmisividad) son conocidos, así como las temperaturas de chapa, vidrio e interior. Por lo tanto, una vez determinados todos los términos de calor se procede a realizar un balance para así conocer el calor que debe aportar o extraer el sistema de climatización del edificio. La expresión del balance de calor mostrado en la [Figura 25](#) es la que se muestra a continuación:

$$Q_{HVAC} + Q_{CO} + Q_{SW} + Q_{LW} = 0 \quad \Rightarrow \quad Q_{HVAC} = -Q_{CO} - Q_{SW} - Q_{LW}$$

5.4.3. Consumo y ahorro de iluminación, calefacción y aire acondicionado.

Como se ha comentado en apartados anteriores, la estimación que el ejecutable realiza del consumo y ahorro del edificio en iluminación, calefacción y aire acondicionado proviene de datos experimentales tomados en Bilbao y en Sevilla. Se considera que estas dos ubicaciones son representativas de una zona climática fría y húmeda, y otra seca y calurosa [6].

La estimación considera cuatro parámetros de entrada: la ubicación geográfica (Bilbao o Sevilla), el ratio de perforación de la chapa, el material de la chapa y la anchura de la cámara. Se cuenta con unas tablas de datos en los que están tabulados, para unas cuantas combinaciones de dichos parámetros, los valores de consumo y ahorro que presenta un edificio de fachada ventilada de chapa perforada.

Para poder implementar dichos datos en el ejecutable se definen una serie de regresiones polinómicas de los datos, las cuales se implementan directamente en el código de VB.NET. Para la obtención de dichas regresiones o líneas de tendencia se ha utilizado el software de cálculo Microsoft Office Excel, el cual permite crear automáticamente este tipo de regresiones a partir de los datos tabulados.

Las expresiones implementadas en el ejecutable consideran como variable el ratio de perforación de la chapa, es decir, devuelven el valor del consumo o el ahorro en función del ratio de perforación de la chapa. Por ello, se ha definido una regresión para cada combinación de material y anchura de cámara en cada ciudad. Por ejemplo, existe una función de regresión que representa, en función del ratio de perforación, el consumo en Bilbao para una anchura de cámara de 5cm y chapa de aluminium anodized; otra que representa el consumo en Bilbao para una anchura de 30cm y con el mismo material. Así sucesivamente con todas las combinaciones posibles. Este cálculo está implementado en la función *Calcular_Ilu_Heat_AC*.

6. EJEMPLO DE APLICACIÓN

En este apartado se plantea un hipotético caso de estudio y se expone, paso por paso, la resolución y análisis de dicho mediante el uso de la aplicación desarrollada. Tras exponer el enunciado del caso, el apartado se estructura en dos partes: introducción de datos y obtención de resultados.

Se plantea el estudio del comportamiento térmico de un edificio con fachada de doble piel de chapa perforada situado en Bilbao. Se desea comparar la demanda energética del sistema de acondicionamiento térmico del edificio (*HVAC*) en invierno y en verano, en función del ratio de perforación de la chapa. Además, se desea también analizar el consumo que dicho edificio presenta a lo largo del año en iluminación, calefacción y aire acondicionado. De esta manera, se desea estimar el ratio de perforación que permita el mayor ahorro posible. Se desea que la cámara de aire pueda ser accesible para los operarios de mantenimiento del edificio, por lo que se ha optado por poner una separación de 30cm entre la chapa y el vidrio. Por último, se desea seleccionar el material y color de la chapa metálica que mejor se ajuste a las condiciones climatológicas de Bilbao.

6.1. Introducción de los datos

En este subapartado se explica, paso por paso, como introducir todas las condiciones solicitadas en el ejecutable.

En primer lugar, lo más aconsejable es crear los proyectos de trabajo que se consideren oportunos y/o necesarios. En este caso en particular se pide comparar la demanda energética del edificio en verano y en invierno, por lo que se ha optado por crear dos proyectos: *Bilbao_verano* y *Bilbao_invierno*. Para ello, en la pestaña de *Proyecto* de la ventana principal se pulsa el botón de *Nuevo*. Al hacerlo, se abrirá un cuadro de diálogo que solicita el nombre del proyecto (ver [Figura 26](#)). Una vez introducido se hace click en *Aceptar* y el nuevo proyecto aparece en el cuadro de resumen de contenido.

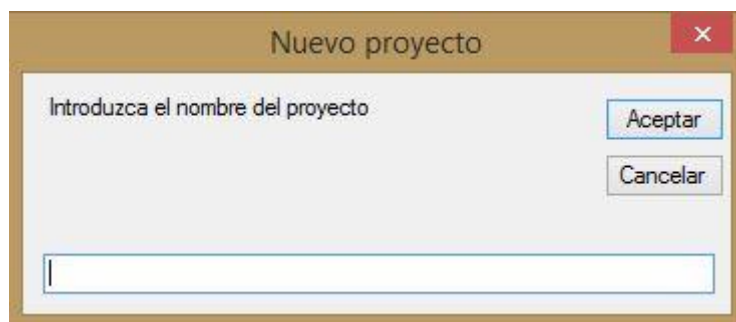


Figura 26: Cuadro de diálogo de introducción del nombre del proyecto

Una vez creados los dos proyectos, el cuadro de resumen muestra el siguiente aspecto:

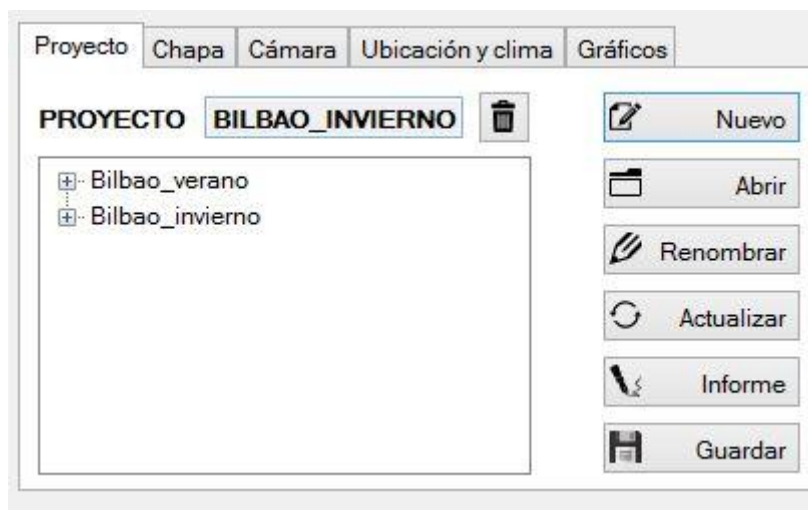


Figura 27: Resumen de proyectos

Con los proyectos ya creados, se procede ahora a determinar las condiciones de cada uno de ellos. Para ello, primero se deben introducir todos los parámetros que definan las condiciones del proyecto.

En primer lugar, en la pestaña *Chapa* se debe seleccionar el material que conforma la chapa. En este caso en particular se pide realizar una comparativa del comportamiento de la fachada para diferentes materiales, por lo que de primeras se mantendrá la opción seleccionada por defecto: *Aluminium anodized* (ver Figura 28). Si bien es cierto que podría generarse un proyecto para cada material, en este caso no resulta necesario puesto que, como se verá más adelante, existe la opción de comparar los cuatro materiales de chapa más comunes al mismo tiempo.

El otro parámetro que permite establecer la pestaña *Chapa* es el ratio de perforación. En este caso, se trata del parámetro que se desea optimizar. Si bien para ello se recurrirá a las soluciones gráficas, en una primera aproximación se le asignará un valor medio: 50%. Para ello, se manipula la flecha superior del cuadro en el que se muestra el valor, hasta que este muestre el valor de 50. Finalmente, *Chapa* mostrará el siguiente aspecto:

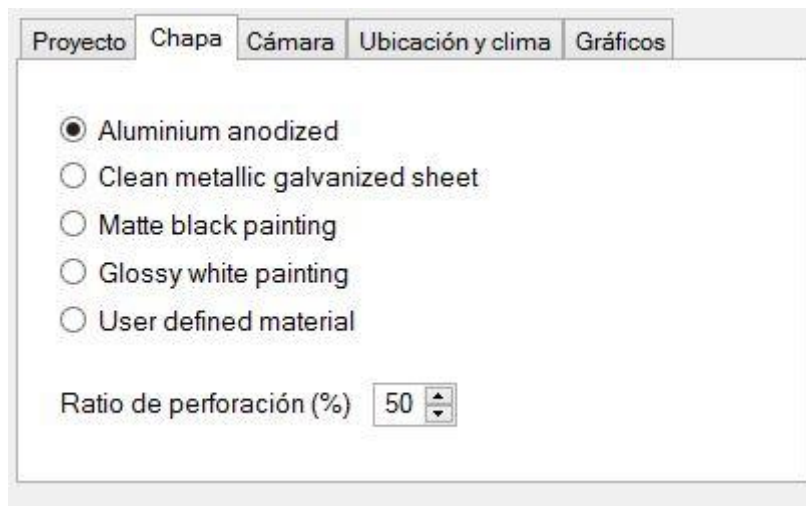


Figura 28: Datos de la chapa

Si se quisiera analizar un material distinto a los que ofrece el ejecutable, se debe seleccionar la opción *User defined material*. Al hacerlo, inmediatamente se mostrará en pantalla la ventana de la [Figura 29](#), la cual permite introducir de forma manual los valores de absorptividad y emisividad que se correspondan con dicho material.

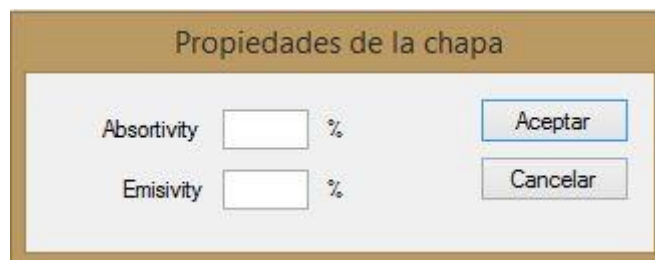


Figura 29: Introducción manual de los parámetros del material de la chapa

Tras introducir los parámetros de la chapa, la pestaña *Cámara* permite determinar la anchura de la cámara. Para ello, el ejecutable permite dos opciones: *5cm* o *30cm*. En este caso en concreto, se especifica que la separación debe ser de *30cm*, que es la opción seleccionada por defecto. Por ello, no se realiza ninguna modificación en esta pestaña.



Figura 30: Introducción de la anchura de la cámara

Una vez especificados los datos de la chapa y la anchura de la cámara, se procede a determinar las condiciones climatológicas que se desean modelar. Para ello la pestaña *Ubicación y clima* ofrece dos posibilidades: seleccionar una de las ubicaciones disponibles o introducir manualmente los datos de irradiación solar y temperatura ambiente. Esta última opción se lleva a cabo seleccionando la opción *Introducir datos concretos de irradiación y temperatura* de la parte inferior de la pestaña. Al marcar dicha opción, se abre el cuadro de diálogo de la [Figura 31](#) que permite la introducción manual de los valores de dichos parámetros.



Figura 31: Introducción manual de las condiciones ambientales

En este caso se pide una ubicación concreta: Bilbao. Por ello, se seleccionará Bilbao entre las ubicaciones disponibles en el ejecutable. Para ello, antes debe determinarse la región en la que se ubica. En este caso, Bilbao está en España por lo que como región se selecciona España. Al hacerlo, si se despliega el menú de ciudades se observa que ofrece cuatro opciones diferentes dentro de España, de la cual se ha de seleccionar Bilbao (ver [Figura 32](#)).

Proyecto Chapa Cámara Ubicación y clima Gráficos

España

Bilbao

Enero

Seleccione el tipo de día

Introducir datos concretos de irradiancia y temperatura

Figura 32: Determinar la ubicación

Se puede observar que, al seleccionar Bilbao, el cuadro de debajo se posiciona automáticamente en enero. Como se están introduciendo las condiciones de *Bilbao_verano*, se cambia esta opción para seleccionar julio; el cual se considera para este ejemplo como mes representativo del verano. Como se van a comparar el verano y el invierno, carece de sentido determinar el tipo de día, por lo que dicha opción se deja sin seleccionar. Finalmente, el aspecto que presenta la pestaña *Ubicación y clima* es el siguiente:

Proyecto Chapa Cámara Ubicación y clima Gráficos

España

Bilbao

Julio

Seleccione el tipo de día

Introducir datos concretos de irradiancia y temperatura

Figura 33: Introducción de datos de clima veraniego en Bilbao

Los datos requeridos por *Bilbao_verano* ya están introducidos. El último paso es agregar dichos parámetros al propio proyecto *Bilbao_verano*. Para ello, se vuelve a la pestaña *Proyecto*, se selecciona¹³ *Bilbao_verano* y se pulsa el botón de actualizar. Al hacerlo, si se despliega el proyecto se ve como los datos se corresponden con los introducidos (ver [Figura 34](#)).

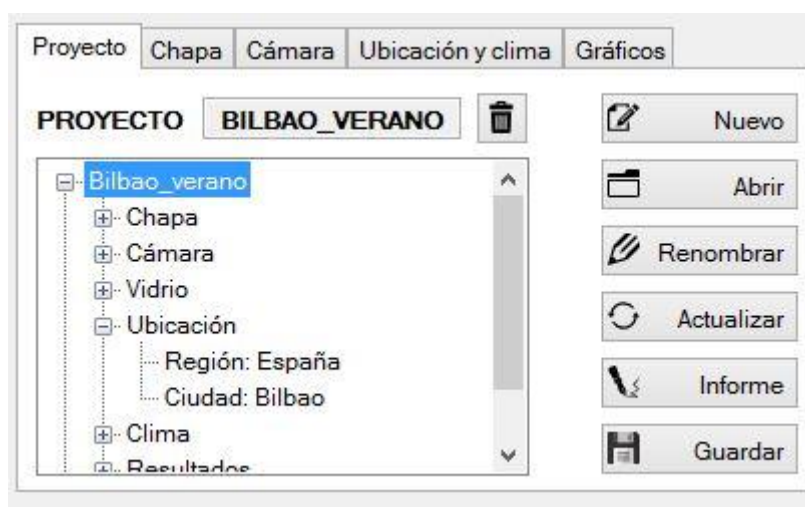


Figura 34: Ejemplo de datos de proyecto actualizados

Una vez introducidos los datos de *Bilbao_verano*, se guarda el proyecto para evitar que se pierda. Al pulsar el botón *Guardar*, se abre un cuadro de diálogo que permite seleccionar la carpeta en la que se desean guardar los datos. Dicho cuadro de diálogo pide, además, que se le de un nombre al fichero del proyecto. El fichero de datos generado puede ser cargado al programa como proyecto mediante el botón *Abrir*. Es MUY IMPORTANTE asegurarse de que se ha hecho doble click sobre el proyecto que se desea guardar antes de guardarlo, puesto que de lo contrario el ejecutable guardará los datos de ejecución del instante en que se haya pulsado el botón de *Guardar*.

Tras guardar el proyecto *Bilbao_verano*, se introducen los datos de *Bilbao_invierno*, se actualizan los datos y se guarda junto con *Bilbao_verano*. La única diferencia entre uno y otro es el mes, ya que en el caso de invierno el mes seleccionado en este ejemplo como representativo es febrero.

6.2. Obtención de los resultados

¹³ Para seleccionar el proyecto se debe hacer click directamente sobre su nombre. Si simplemente se expande sin clicar sobre el nombre no se selecciona.

En este subapartado se expone el método para visualizar, generar y guardar los resultados obtenidos para las condiciones introducidas al ejecutable. Se comienza con una explicación de dónde buscar y cómo almacenar los resultados numéricos obtenidos por el ejecutable. A continuación, se muestra el modo de obtención de gráficos y su almacenamiento.

6.2.1. Resultados numéricos

Con los datos de partida de los proyectos *Bilbao_verano* y *Bilbao_invierno* el ejecutable obtiene, numéricamente, los siguientes resultados:

- Temperaturas medias de la chapa, el vidrio y la cámara.
- Demanda de HVAC (en W/m²) del edificio bajo las condiciones especificadas.
- Consumo anual (en kWh), con el ratio de perforación y el material escogidos, de iluminación, calefacción y aire acondicionado.
- Ahorro anual (en %) de iluminación y aire acondicionado e incremento del consumo de calefacción (en %) frente a una fachada sin chapa perforada.

En el propio ejecutable, los resultados están disponibles siempre de dos maneras. La primera de ellas, la más compacta, es el último nodo desplegable del resumen del proyecto, de nombre *Resultados* (ver [Figura 35](#)). Si se expande, se ve que dichos resultados se encuentran clasificados en el mismo orden que el expuesto en este apartado.



Figura 35: Resultados en el resumen

De esta manera, los resultados se hace accesibles de forma rápida y sencilla, además de estar ordenados de forma esquemática y compacta. No obstante, existe otra

ubicación para cada uno de estos resultados en el ejecutable, buscando hacer que su significado sea más visual. Además, en caso de variar las condiciones, los resultados mostrados en el resumen de proyecto se mantienen estáticos si éste no se actualiza. Es decir, dichos resultados se corresponden exclusivamente a las condiciones de dicho proyecto.

Si se observa la zona derecha de la ventana principal, en ella hay un esquema constructivo de la fachada sobre el que se indican las temperaturas tanto de la chapa, el vidrio y la cámara, como la demanda de HVAC (ver [Figura 36](#)). Los valores ahí mostrados, en caso de que la casilla *Automático* esté seleccionada, se corresponden con las condiciones actuales del proyecto. Esto resulta extremadamente útil si, por ejemplo, se desea conocer el efecto de aumentar la temperatura ambiente sobre la chapa. Para ello, basta con arrastrar el marcador de la barra que representa dicha temperatura y el ejecutable, de forma automática, recalcula todos los valores. Para las condiciones de *Bilbao_invierno*, por ejemplo, dichas temperaturas son las que se muestran en la [Figura 36](#). En ella se observan, al mismo tiempo, las temperaturas del resumen del proyecto y del esquema constructivo.

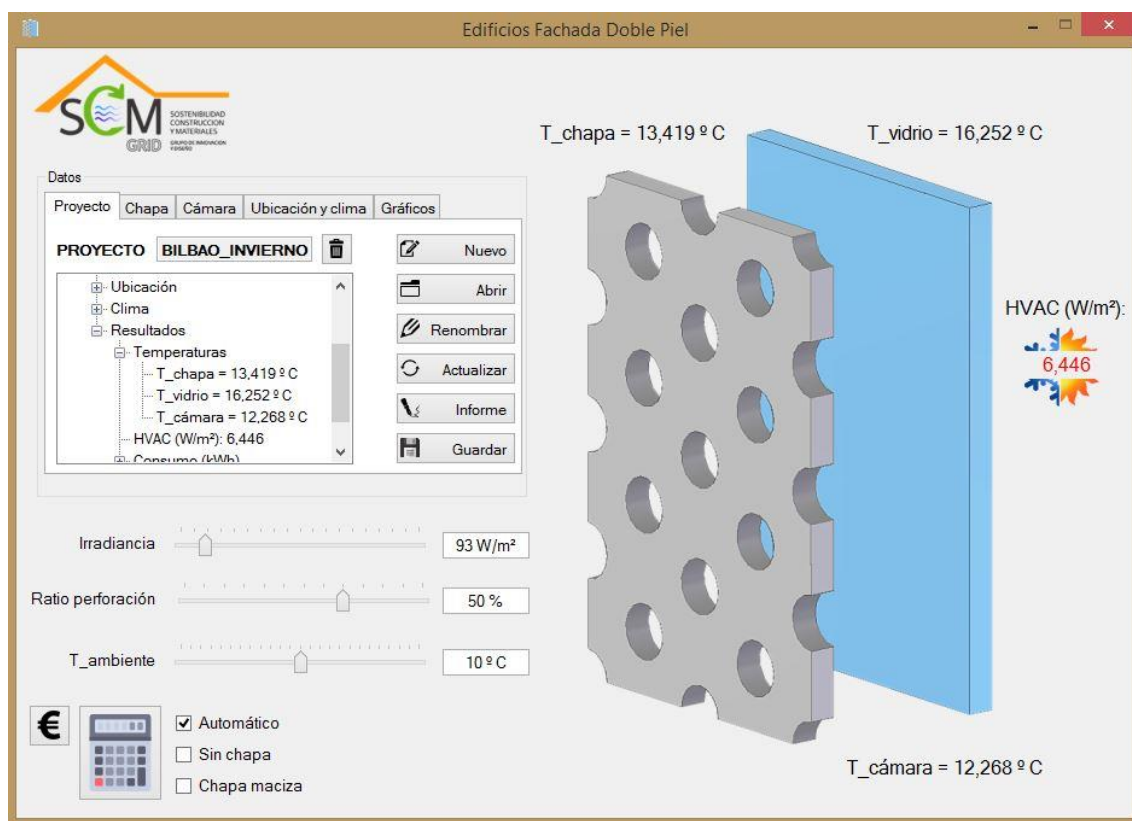


Figura 36: Temperaturas coincidentes en el resumen y en el esquema constructivo

Por el contrario, si se observa la [Figura 37](#), se comprueba como tras variar el valor de la temperatura ambiente los resultados del proyecto permanecen inalterados y los del esquema constructivo han variado.

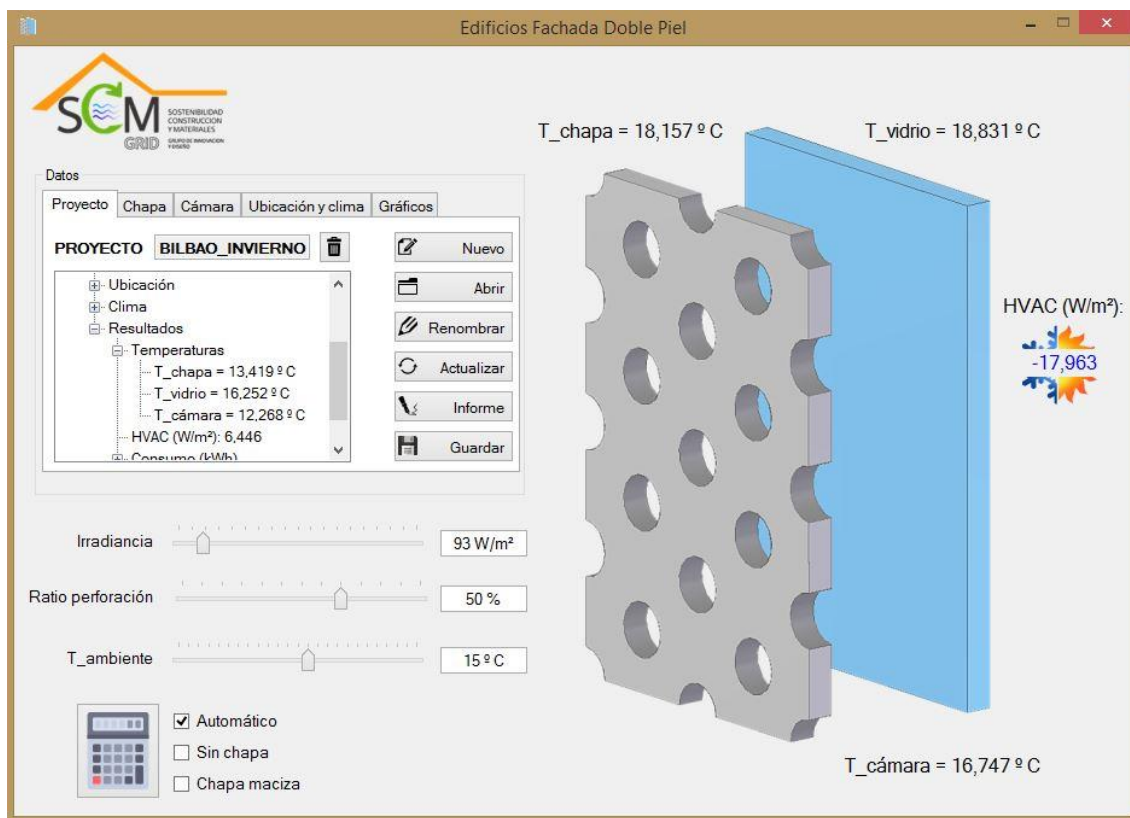
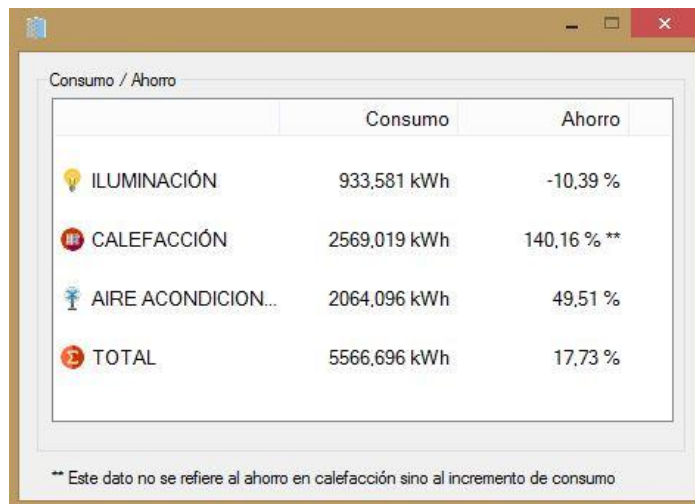


Figura 37: Temperaturas diferentes en el resumen y en el esquema constructivo

Los datos de consumo y ahorro son accesibles en el ejecutable pulsando sobre el botón con el icono € de la zona inferior derecha. Cabe mencionar que dicho icono solo es accesible para condiciones climáticas de Bilbao o Sevilla, puesto que como se mencionó en el apartado [5.4.3](#) los resultados de consumo y ahorro son de origen experimental y solo están disponibles para esas dos ciudades.

Al pulsarlo, se muestra en pantalla la ventana de la [Figura 38](#). En ella, se muestra una tabla que contiene los resultados. Al igual que las temperaturas, dicha los resultados mostrados en dicha tabla se actualizan de forma automática al variar el ratio de perforación¹⁴. En la [Figura 38](#) se muestran los datos de consumo y ahorro para *Bilbao*.

¹⁴ En caso de alterar alguna de las condiciones ambientales el contenido de esta tabla desaparece, ya que un cambio en el ambiente implica un cambio de ubicación.



	Consumo	Ahorro
ILUMINACIÓN	933,581 kWh	-10,39 %
CALEFACCIÓN	2569,019 kWh	140,16 % **
AIRE ACONDICION...	2064,096 kWh	49,51 %
TOTAL	5566,696 kWh	17,73 %

** Este dato no se refiere al ahorro en calefacción sino al incremento de consumo

Figura 38: Consumo y ahorro anuales de Bilbao

Una vez realizados los cálculos, el ejecutable permite generar un informe en el que se resumen tanto los parámetros de entrada empleados como los resultados obtenidos. Dicho informe se crea en formato Microsoft Office Word y se puede guardar donde se considere, y en el formato que se desee. Un ejemplo de informe generado a partir del ejecutable, el de el caso de *Bilbao_verano*, aparece en el [Anexo II](#).

6.2.2. Representación gráfica de los resultados

La obtención numérica de los resultados no es la única herramienta que posee el ejecutable. Además, permite representar gráficamente los resultados. Para ello, cuenta con la pestaña gráficos, que se muestra en la [Figura 39](#).



Figura 39: Pestaña de Gráficos

En este caso concreto, se pide analizar la demanda de HVAC tanto en verano como en invierno. En primer lugar, se va a realizar el gráfico de verano. Para ello, el primer paso es volver a la pestaña de *Proyecto* y hacer doble click sobre *Bilbao_verano* para asegurar que los datos que maneja el ejecutable son los de verano. Después, volviendo a la ventana de *Gráficos*, se selecciona la variable *perf_ratio* en el eje X y la variable *HVAC* en el eje Y. De esta manera, se le está señalando al ejecutable cuales son los parámetros que se desea graficar.

Al seleccionar *perf_ratio*, los valores de *x_min* y *x_max* se actualizan al 10% y 70% respectivamente. Esto se debe a que dichos valores son el máximo y el mínimo que permite el ejecutable para el ratio de perforación.

Generalmente, la variación de HVAC se presenta en gráficos de barras. Por lo tanto, a fin de evitar que el gráfico resultante esté muy saturado, se aumenta *Intervalo* a 2. Esto hace que, a la hora de graficar, las barras no se muestren para cada valor del ratio de perforación entre 10% y 70%, sino que se en intervalos del 2%.

Una vez seleccionado todo, la pestaña *Gráficos* muestra el siguiente aspecto:



Figura 40: Datos para graficar p vs HVAC en Bilbao_verano

Para crear el gráfico, se hace click sobre el botón con el icono del gráfico de barras. El resultado es el que se muestra en la [Figura 41](#).

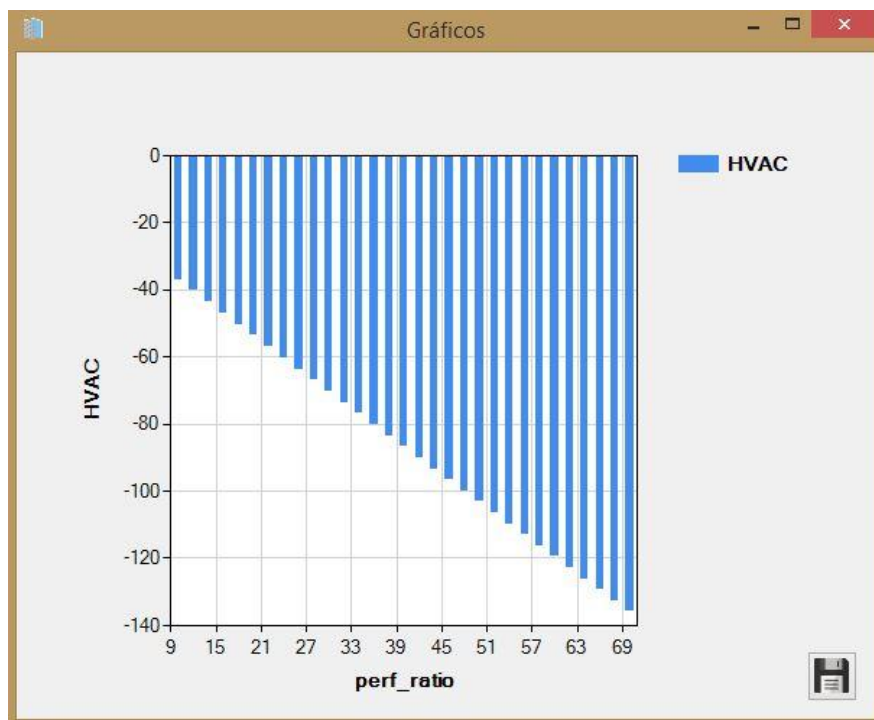


Figura 41: Gráfico de p vs HVAC para Bilbao_verano

En la [Figura 41](#) se observa como, en verano, la demanda de HVAC decrece con el ratio de perforación. Esto se debe a que, al aumentar el porcentaje de perforaciones, una mayor cantidad de radiación solar incide directamente sobre el vidrio, aumentando así el término de calor por radiación directa que accede al edificio.

Si se realiza el mismo gráfico para las condiciones invernales, el resultado es que mostrado en la [Figura 42](#).

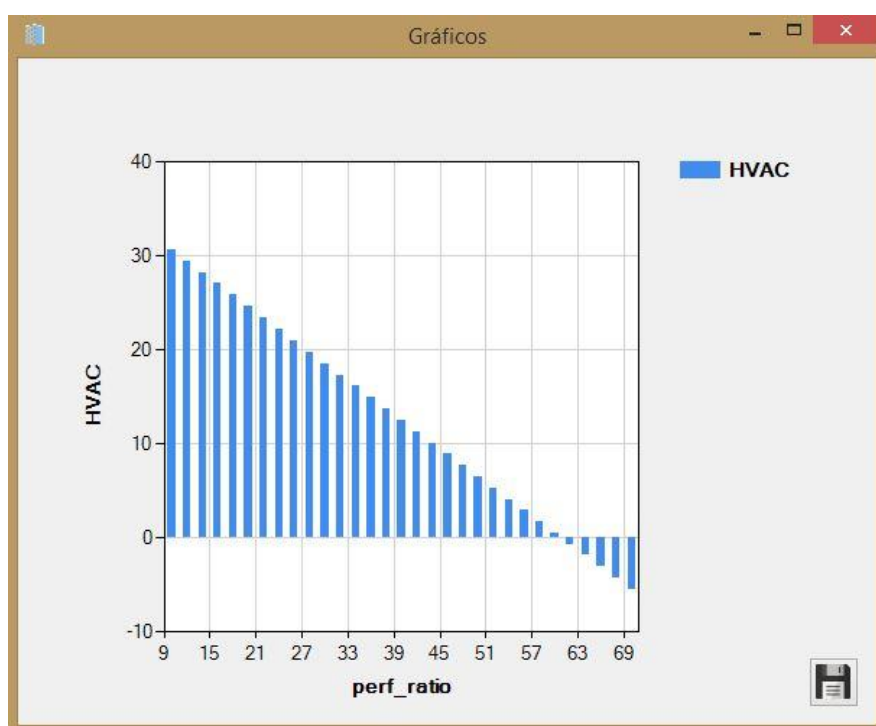


Figura 42: Gráfico de p vs HVAC para *Bilbao_invierno*

En el caso de la [Figura 42](#), se observa como en invierno la demanda de HVAC disminuye a medida que aumenta el ratio de perforación hasta cierto valor mínimo, tras el cual comienza a aumentar de nuevo. Esto se debe al aprovechamiento que se hace de la radiación solar directa que incide sobre el vidrio. Lo que en verano supone un aporte de calor indeseado, en invierno puede suponer un ahorro en calefacción.

Si se desea comparar el consumo de HVAC en función de los diferentes materiales de chapa disponibles, se selecciona la variable *perf_ratio* en la pestaña de *Gráficos* (ver [Figura 40](#)) pero, en lugar de pulsar el botón con el icono de las barras, se pulsa el botón de más a la derecha. Con ello se muestra en pantalla un cuadro de diálogo que permite seleccionar diferentes opciones (ver [Figura 43](#)).

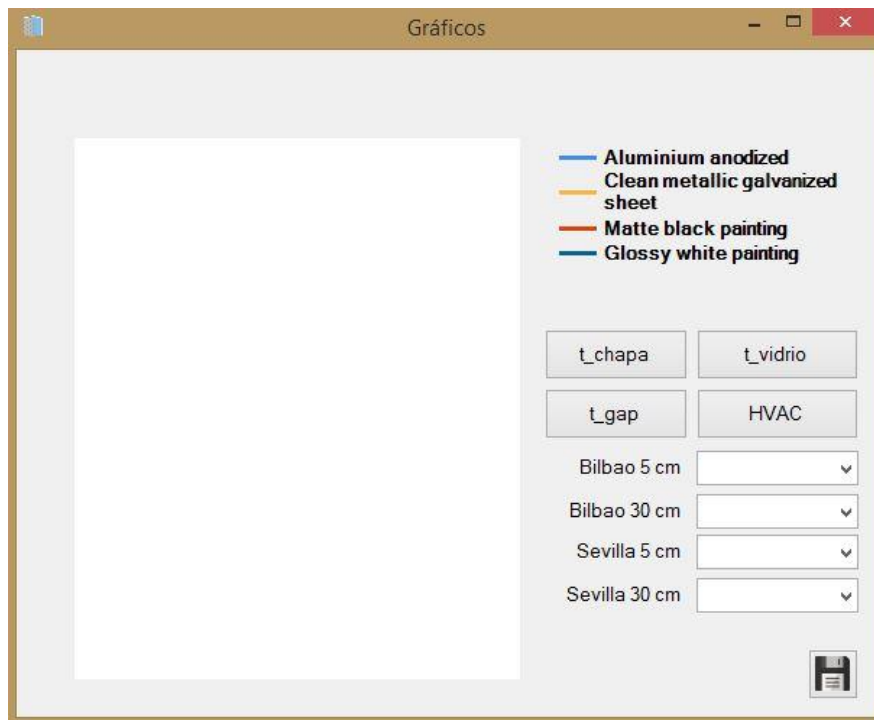


Figura 43: Ventana de gráficos para comparar materiales

La ventana de la [Figura 43](#) permite realizar numerosas comparativas. En primer lugar, siguiendo con el planteamiento del ejemplo, se puede graficar el consumo de HVAC tanto en verano como en invierno para los diferentes materiales (ver [Figura 44](#)). Para ello, se pulsa sobre el botón *HVAC*.

Si se quisiera comparar el consumo y ahorro anuales de iluminación, calefacción y aire acondicionado para cada uno de los materiales, se podría graficar dicha comparativa mediante los menús desplegables. Cada uno de ellos permite las opciones de *Iluminación*, *Calefacción* y *Aire acondicionado*, de forma que al seleccionar, por ejemplo, el consumo y ahorro de aire acondicionado en Bilbao para el caso de una fachada de 30 cm, se obtiene la [Figura 45](#).

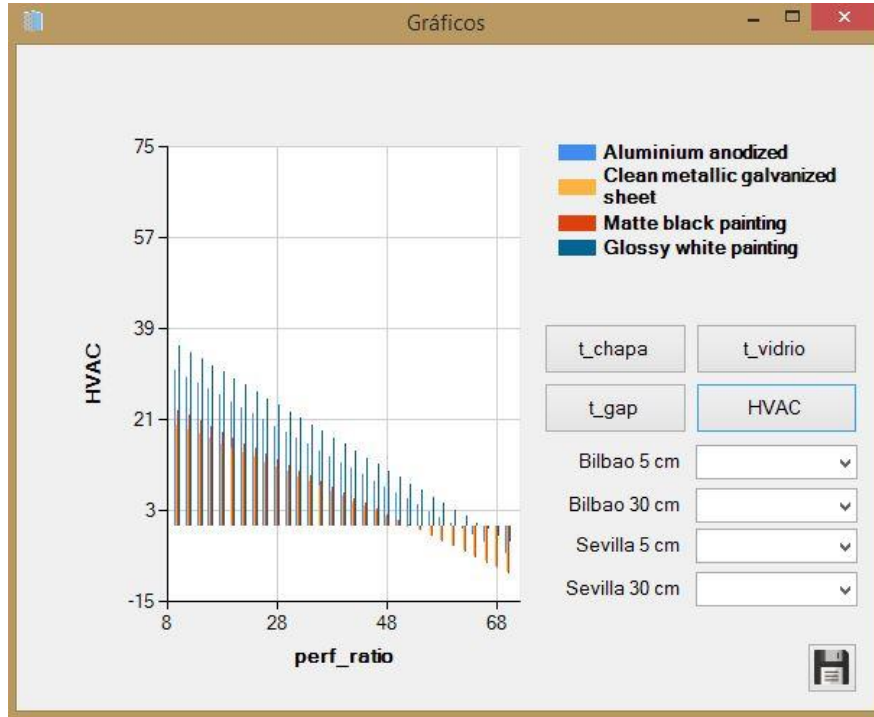


Figura 44: Comparativa de materiales en la demanda de HVAC para *Bilbao_invierno*

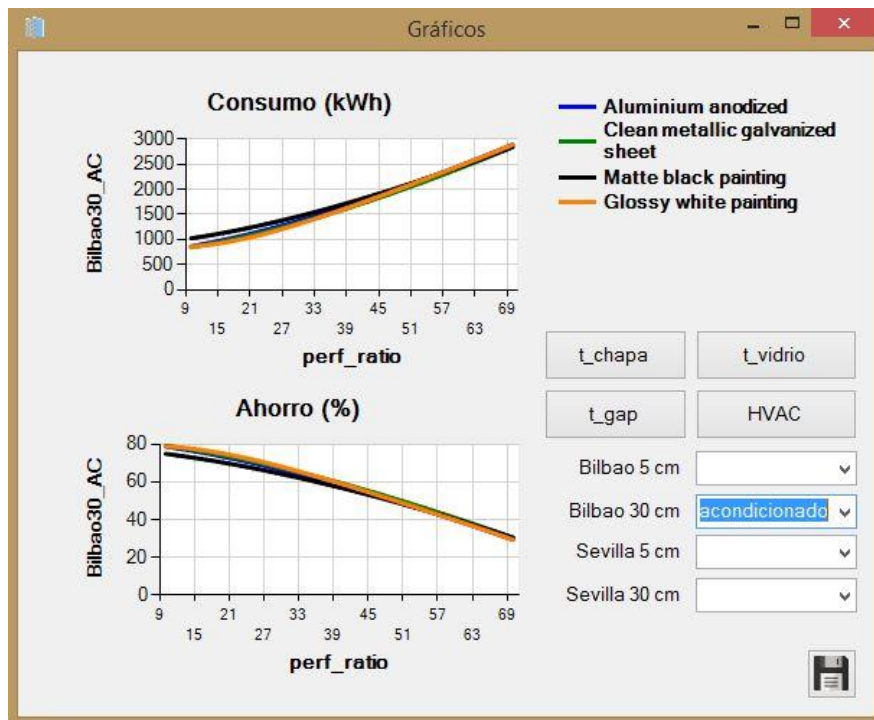


Figura 45: Comparativa de materiales en el consumo de aire acondicionado para Bilbao 30 cm



BILBOKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE BILBAO

Por último, cabe recordar que el botón de la zona inferior derecha convierte el gráfico actual de dicha ventana en un archivo con formato de imagen (.png), el cual permite guardar en la ubicación que el usuario especifique.

7. PROPUESTAS DE MEJORA

En este apartado se exponen una serie de mejoras que, en un futuro desarrollo, podrían ser implementadas y enriquecer así la solución desarrollada.

- **Optimización del código:** las funciones y procedimientos con los que funciona el ejecutable, si bien siguen una lógica que funciona, no son demasiado versátiles. Esto se debe a que su estructura, en cuanto a los parámetros de entrada y salida con los que operan, ha sido diseñada con el fin de que cumplieran su utilidad por lo que, de cara a un futuro desarrollo del ejecutable, sería conveniente revisar el código y escribirlo de forma más amoldable a nuevas funcionalidades.
- **Función de derivación numérica:** el método de Newton-Raphson, el cual es aplicado por el ejecutable para obtener las temperaturas de la chapa y el vidrio, requiere el cálculo de la matriz Jacobiana del sistema. Actualmente, dicha matriz ha sido calculada de forma analítica por medio del software Mathematica®, por lo que cualquier cambio en el sistema de ecuación implica la necesidad de volver a derivar todo el sistema analíticamente. Este paso se puede solventar introduciendo una función que, de forma numérica, sea capaz de realizar el cálculo de dicha matriz Jacobiana.
- **Consideración del viento:** en el modelo planteado el coeficiente de la chapa con el exterior se corresponde con el valor recomendado para condiciones de convección natural. Por lo tanto, una manera de ampliar el alcance del ejecutable es modificar dicho coeficiente de forma que la velocidad del viento sea considerada, y pueda así el ejecutable operar con condiciones de convección forzada.
- **Consideración de la lluvia:** así como el coeficiente de convección externa actual considera una velocidad del viento nula, considera también un ambiente seco. Por lo tanto, una manera de ampliar las capacidades de este ejecutable es modificar dicho coeficiente para que tenga en cuenta también la humedad de la fachada o del ambiente.
- **Interoperabilidad con Excel:** el ejecutable cuenta con datos climatológicos mensuales medios de una serie de ciudades del mundo. Además, permite la introducción manual de datos de irradiancia y temperatura ambiente. Si bien esto permite considerar cualquier condición para la que el ejecutable ha sido desarrollado, presenta ciertas limitaciones en cuanto al análisis de períodos temporales como puede ser el transcurso de un día. Por ello, una mejora muy interesante es añadir la opción de seleccionar una secuencia de datos de irradiancia y temperatura ambiente desde una tabla de Excel externa, y obtener directamente una tabla con los resultados para cada una de dichas condiciones.

8. PLANIFICACIÓN

En esta sección se establecen la definición de tareas y el diagrama de Gantt de este proyecto.

8.1. Definición de tareas

Este proyecto ha sido dividido en cuatro fases, las cuales a su vez se han dividido en tareas como se muestra a continuación.

Fase 1: Documentación

Tarea 1.1: Documentación sobre fachadas ventiladas y su modelado térmico.

- Descripción: búsqueda y revisión de la bibliografía referente al funcionamiento térmico de las fachadas de doble piel, poniendo especial interés sobre las de chapa perforada y, más en concreto, sobre [4] y [6].
- Recursos humanos y carga de trabajo: ingeniero junior (10 h)
- Recursos técnicos: ordenador con acceso a internet.
- Duración: 3 semanas.

Tarea 1.2: Selección de alternativa y aprendizaje del lenguaje de programación escogido.

- Descripción: búsqueda, análisis y selección de alternativas. Familiarización y aprendizaje del lenguaje de programación seleccionado.
- Recursos humanos y carga de trabajo: ingeniero sénior (2 h), ingeniero junior (30 h).
- Recursos técnicos: ordenador con acceso a internet, biblioteca de la EIB.
- Duración: 5 semanas.

Fase 2: Desarrollo del ejecutable

Tarea 2.1: Diseño de la interfaz gráfica.

- Descripción: diseño y desarrollo de la interfaz gráfica de usuario del ejecutable.
- Recursos humanos y carga de trabajo: ingeniero sénior (3 h), ingeniero junior (10 h).
- Recursos técnicos: ordenador con acceso a internet, entorno de desarrollo.
- Duración: 1 semana.

Tarea 2.2: Desarrollo de la estructura interna del ejecutable.

- Descripción: desarrollo e implementación del código del ejecutable.
- Recursos humanos y carga de trabajo: ingeniero junior (30 h).
- Recursos técnicos: ordenador con acceso a internet, entorno de desarrollo, manuales de programación, licencia de Matlab[®], licencia de Wolfram Mathematica[®].
- Duración: 5 semanas.

Fase 3: Validación de la solución

Tarea 3.1: Validación de los resultados del ejecutable.

- Descripción: validación de los resultados numéricos obtenidos con el ejecutable comparándolos con el modelo de Matlab[®] desarrollado por *Jesús Mari et al.* en [4].
- Recursos humanos y carga de trabajo: ingeniero sénior (2 h), ingeniero junior (10 h).
- Recursos técnicos: ordenador con acceso a internet, licencia de Matlab[®], modelo matemático desarrollado por *Jesús Mari et al.* en Matlab[®].
- Duración: 1 semana.

Tarea 3.2: Depuración de errores.

- Descripción: determinación y corrección de errores en tiempo de ejecución del ejecutable.
- Recursos humanos y carga de trabajo: ingeniero junior (15 h).
- Recursos técnicos: ordenador, ejecutable, entorno de desarrollo.
- Duración: 3 semana.

Fase 4: Documentación del resultado

Tarea 4.1: Redacción de la memoria del proyecto.

- Descripción: redacción de un documento en el que se describan el ejecutable y su desarrollo en detalle.
- Recursos humanos y carga de trabajo: ingeniero junior (20 h).
- Recursos técnicos: ordenador con Microsoft Office.
- Duración: 3 semanas.

8.2. Diagrama de Gantt

Año	2018																						
Mes	Enero			Febrero					Marzo					Abril					Mayo			Junio	
Semana	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23		
Fase 1																							
Tarea 1.1																							
Tarea 1.2																							
Fase 2																							
Tarea 2.1																							
Tarea 2.2																							
Fase 3																							
Tarea 3.1																							
Tarea 3.2																							
Fase 4																							
Tarea 4.1																							

Figura 46: Diagrama de Gantt

9. CONCLUSIONES

Se ha desarrollado un ejecutable que modela el comportamiento térmico de una fachada de doble piel de chapa perforada. El ejecutable permite la introducción, bien manual o bien desde un fichero de datos externo, de los parámetros ambientales de irradiancia solar y temperatura ambiente; así como de parámetros constructivos de la fachada: material, color y ratio de perforación de la chapa y anchura de la cámara. Además, ofrece la posibilidad de introducir dichos parámetros ambientales en forma de ubicaciones geográficas concretas (ciudades) en momentos temporales concretos (meses). Con todos los parámetros de entrada el ejecutable es capaz de calcular, las temperaturas de la chapa, el vidrio y la cámara de la fachada (consideradas constantes); la demanda de HVAC que presenta el edificio y, en el caso de haber seleccionado como ciudad Bilbao o Sevilla, el consumo y ahorro anuales de iluminación, calefacción y aire acondicionado.

El modelo matemático descrito, en el cual se basan los cálculos del ejecutable, ha sido validado por medio de mediciones experimentales realizadas en la fachada del Edificio II de la EIB (Escuela de Ingeniería de Bilbao). A su vez, los resultados del ejecutable han sido validados por medio de dos funciones de Matlab[®] desarrolladas por *Jesús Mari et al.* en las que está implementado el modelo matemático.

El ejecutable permite generar, de forma automática, un informe en formato Word (.docx) en el que se resumen tanto los parámetros de entrada introducidos como los resultados obtenidos a partir de dichos parámetros. Además, permite almacenar las variables utilizadas en ficheros de datos de forma que pueden volver a ser cargadas por el propio ejecutable. En cuanto a la representación gráfica, el ejecutable permite graficar los resultados en función de los parámetros de entrada, con la opción añadida de guardar los gráficos generados en un archivo con formato de imagen (.png).

El ejecutable desarrollado en este proyecto es una herramienta visual y sencilla que hace accesibles cálculos complejos de ingeniería térmica a usuarios con escaso o nulo dominio de la materia. Permite una fácil introducción de parámetros que, combinadas con una representación esquemática y visual de los resultados, facilitan la comprensión de los fenómenos de transferencia de calor que se dan en este tipo de fachadas. Por ello, supone una herramienta útil a la hora de justificar tomas de decisiones como, por ejemplo, la selección de un material y no otro para la chapa. A todo esto hay que añadir su posible uso en el ámbito académico. Por su sencillez, podría ser utilizado tanto por estudiantes de grado como de postgrado a modo de apoyo en su proceso de comprensión de la transferencia de calor en fachadas ventiladas de chapa perforada. Suponiendo para ellos una herramienta de cálculo básica y sencilla sobre la que realizar pruebas y diferentes combinaciones para analizar los resultados.

ANEXO I: CÓDIGO DEL EJECUTABLE.

FormPrincipal

```

Imports System.Math
Imports Word = Microsoft.Office.Interop.Word

Public Class FormPrincipal
    Public titulo As String

    Private Sub FormPrincipal_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        optAlumnAnod.Checked = True
        checkDatosPersonalizados.Checked = False
        p = Cdbl(numPerfRate.Value) / 100
        Ir_sol = 300
        t_amb = 273 + 15
        opt30cm.Checked = True
        material = optAlumnAnod.Text
        separacion = 30
        checkAutomatico.Checked = True
        checkLeyendaGraficos.Checked = True
        titulo = ""
        lblUnidadesXmax.Text = ""
        lblUnidadesXmin.Text = ""
        lblUnidadesIntervalo.Text = ""
        btnConsumoAhorro.Visible = False
        numIntervalo.Value = 1
    End Sub

    Private Sub optAlumnAnod_CheckedChanged(sender As Object, e As EventArgs) Handles optAlumnAnod.CheckedChanged

        If optAlumnAnod.Checked = True Then
            alpha_solar_chapa = 0.5
            epsilon_IR_chapa = 0.84
            material = optAlumnAnod.Text

            If checkAutomatico.Checked = True Then
                Hacer_calculos()
            End If
        End If
    End Sub

    Private Sub optGalvanSteel_CheckedChanged_1(sender As Object, e As EventArgs) Handles optGalvanSteel.CheckedChanged

        If optGalvanSteel.Checked = True Then
  
```

```

    alpha_solar_chapa = 0.65
    epsilon_IR_chapa = 0.13
    material = optGalvanSteel.Text

    If checkAutomatico.Checked = True Then
      Hacer_calculos()
    End If

  End If

End Sub

Private Sub optMatteBlack_CheckedChanged_1(sender As Object, e As EventArgs)
Handles optMatteBlack.CheckedChanged

  If optMatteBlack.Checked = True Then
    alpha_solar_chapa = 0.95
    epsilon_IR_chapa = 0.89
    material = optMatteBlack.Text

    If checkAutomatico.Checked = True Then
      Hacer_calculos()
    End If

  End If

End Sub

Private Sub optGlossyWhite_CheckedChanged_1(sender As Object, e As EventArgs)
Handles optGlossyWhite.CheckedChanged

  If optGlossyWhite.Checked = True Then
    alpha_solar_chapa = 0.25
    epsilon_IR_chapa = 0.89
    material = optGlossyWhite.Text

    If checkAutomatico.Checked = True Then
      Hacer_calculos()
    End If

  End If

End Sub

Private Sub optUserDefined_CheckedChanged_1(sender As Object, e As EventArgs)
Handles optUserDefined.CheckedChanged

  material = optUserDefined.Text

  If optUserDefined.Checked = True Then
    FormUserDefMaterial.ShowDialog()
  End If

```

End Sub

```
Private Sub cboRegión_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles cboRegión.SelectedIndexChanged
```

```
' En función de la región que se escoja, aparecerán distintas opciones de
ciudades para elegir
```

```
checkDatosPersonalizados.Checked = False
```

```
Select Case cboRegión.SelectedItem
```

```
Case "Australia"
```

```

cboCity.Items.Clear()
cboCity.Items.Add("Sidney")
cboCity.Items.Add("Melbourne")

```

```
Case "China"
```

```

cboCity.Items.Clear()
cboCity.Items.Add("Shangai")
cboCity.Items.Add("Beijing")
cboCity.Items.Add("Hong Kong")
cboCity.Items.Add("Guangzhou")
cboCity.Items.Add("Macau")

```

```
Case "Ciudades Africanas"
```

```

cboCity.Items.Clear()
cboCity.Items.Add("Cape Town")
cboCity.Items.Add("Johannesburgo")
cboCity.Items.Add("Douala")
cboCity.Items.Add("Nairobi")

```

```
Case "Ciudades de América Latina"
```

```

cboCity.Items.Clear()
cboCity.Items.Add("Bogotá")
cboCity.Items.Add("Rio De Janeiro")

```

```
Case "Ciudades Árabes"
```

```

cboCity.Items.Clear()
cboCity.Items.Add("Dubai")

```

```
Case "Ciudades Asiáticas"
```

```

cboCity.Items.Clear()
cboCity.Items.Add("Bangkok")
cboCity.Items.Add("Nueva Delhi")
cboCity.Items.Add("Bombay")

```

```
Case "Ciudades Europeas"
```

```

        cboCity.Items.Clear()
        cboCity.Items.Add("París")
        cboCity.Items.Add("Londres")

    Case "Ciudades Nórdicas"

        cboCity.Items.Clear()
        cboCity.Items.Add("Estocolmo")

    Case "EEUU"

        cboCity.Items.Clear()
        cboCity.Items.Add("Nueva York")
        cboCity.Items.Add("Los Ángeles")

    Case "España"

        cboCity.Items.Clear()
        cboCity.Items.Add("Bilbao")
        cboCity.Items.Add("Madrid")
        cboCity.Items.Add("Sevilla")
        cboCity.Items.Add("Valencia")

    End Select

End Sub

Private Sub cboCity_Click(sender As Object, e As EventArgs) Handles
cboCity.Click

    If cboRegión.Text = "Seleccione una región" Then
        MessageBox.Show("No ha seleccionado ninguna región", "Información",
        MessageBoxButtons.OK,
            MessageBoxIcon.Information)
    End If
End Sub

Private Sub checkDatosPersonalizados_CheckedChanged(sender As Object, e As
EventArgs) Handles checkDatosPersonalizados.CheckedChanged

    If checkDatosPersonalizados.Checked = True Then
        FormIrradTamb.ShowDialog()
    End If

End Sub

Private Sub btnCalcular_Click(sender As Object, e As EventArgs) Handles
btnCalcular.Click

    Hacer_calculos()
    If (p <= 0.7) And (p >= 0.1) Then
        Posicionar_barras(Ir_sol, p, t_amb)
    ElseIf p = 0.999 Then
        Posicionar_barras(Ir_sol, 0.7, t_amb)
  
```

```

        txtBarraPerfRate.Text = "100%"
    End If

End Sub

Private Sub numPerfRate_ValueChanged(sender As Object, e As EventArgs)
Handles numPerfRate.ValueChanged
    If checkSinChapa.Checked = True Then
        checkSinChapa.Checked = False
    End If

    If checkChapaMaciza.Checked = True Then
        checkChapaMaciza.Checked = False
    End If

    If numPerfRate.Value > 99 Then
        numPerfRate.Value = 99
    ElseIf numPerfRate.Value < 0 Then
        numPerfRate.Value = 0
    End If

    p = Cdbl(numPerfRate.Value) / 100

    If checkAutomatico.Checked = True Then
        Hacer_calculos()
        Posicionar_barras(Ir_sol, p, t_amb)
    End If

End Sub

Private Sub cboCity_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles cboCity.SelectedIndexChanged

    If Not (cboCity.Text = "Seleccione una ciudad") Then
        ciudad = cboCity.SelectedItem
        Ir_sol = Determinar_Ir_Tamb_Ciudad(ciudad, cboFecha.Text)(0)
        t_amb = Determinar_Ir_Tamb_Ciudad(ciudad, cboFecha.Text)(1)

        If (ciudad = "Bilbao") Or (ciudad = "Sevilla") Then
            cboFecha.SelectedItem = cboFecha.Items(0)
        End If

        If (checkAutomatico.Checked = True) And Not (cboFecha.Text =
"Seleccione un mes") Then
            Hacer_calculos()
            Posicionar_barras(Ir_sol, p, t_amb)
        End If

    End If

End Sub

Private Sub cboCity_TextChanged(sender As Object, e As EventArgs) Handles
cboCity.TextChanged

```



```

    If (cboCity.Text = "Bilbao") Or (cboCity.Text = "Sevilla") Then
      btnConsumoAhorro.Visible = True
    Else
      btnConsumoAhorro.Visible = False
      FormConsumoAhorro.Close()
    End If
  End Sub

  Private Sub opt5cm_CheckedChanged(sender As Object, e As EventArgs) Handles
opt5cm.CheckedChanged
    If opt5cm.Checked = True Then
      pic30cm.Visible = False
      picMaciza.Visible = False
      pic5cm.Visible = True
      separacion = 5
      checkChapaMaciza.Checked = False
      checkSinChapa.Checked = False
    End If

    If checkAutomatico.Checked = True Then
      Hacer_calculos()
      Posicionar_barras(Ir_sol, p, t_amb)
    End If

  End Sub

  Private Sub opt30cm_CheckedChanged(sender As Object, e As EventArgs) Handles
opt30cm.CheckedChanged
    If opt30cm.Checked = True Then
      pic5cm.Visible = False
      picMaciza.Visible = False
      pic30cm.Visible = True
      separacion = 30
      checkChapaMaciza.Checked = False
      checkSinChapa.Checked = False
    End If

    If checkAutomatico.Checked = True Then
      Hacer_calculos()
      Posicionar_barras(Ir_sol, p, t_amb)
    End If

  End Sub

  Private Sub cboFecha_MouseDown(sender As Object, e As EventArgs) Handles
cboFecha.MouseDown
    If cboCity.Text = "Seleccione una ciudad" Then
      MessageBox.Show("No ha seleccionado ninguna ciudad", "Información",
      MessageBoxButtons.OK,
      MessageBoxIcon.Information)
    End If
  End Sub

```

```

Private Sub trackIrradiancia_Scroll(sender As Object, e As EventArgs) Handles
trackIrradiancia.Scroll
    Ir_sol = Cdbl(trackIrradiancia.Value)
    txtBarraIrrad.Text = CStr(trackIrradiancia.Value) & " W/m²"
    FormIrradTamb.numIrradiancia.Value = trackIrradiancia.Value

    If checkAutomatico.Checked = True Then
        Hacer_calculos()

        cboRegión.Text = "Seleccione una región"
        cboCity.Text = "Seleccione una ciudad"
        ciudad = cboCity.Text
        cboFecha.Text = "Seleccione un mes"
    End If

End Sub

Private Sub trackPerfRate_Scroll(sender As Object, e As EventArgs) Handles
trackPerfRate.Scroll
    If checkSinChapa.Checked = True Then
        checkSinChapa.Checked = False
    End If

    If checkChapaMaciza.Checked = True Then
        checkChapaMaciza.Checked = False
    End If

    p = Cdbl(trackPerfRate.Value / 100)
    txtBarraPerfRate.Text = CStr(trackPerfRate.Value) & " %"
    numPerfRate.Value = trackPerfRate.Value

    If checkAutomatico.Checked = True Then
        Hacer_calculos()
    End If

End Sub

Private Sub trackTamb_Scroll(sender As Object, e As EventArgs) Handles
trackTamb.Scroll
    t_amb = Cdbl(trackTamb.Value + 273)
    txtBarraTamb.Text = CStr(trackTamb.Value) & " ° C"
    FormIrradTamb.numTamb.Value = trackTamb.Value

    If checkAutomatico.Checked = True Then
        Hacer_calculos()

        cboRegión.Text = "Seleccione una región"
        cboCity.Text = "Seleccione una ciudad"
        ciudad = cboCity.Text
        cboFecha.Text = "Seleccione un mes"
    End If

End Sub

```

```

Private Sub cboFecha_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles cboFecha.SelectedIndexChanged
    If Not (cboFecha.Text = "Seleccione un mes") Then
        Ir_sol = Determinar_Ir_Tamb_Ciudad(cboCity.Text,  cboFecha.Text)(0)
' El numero entre paréntesis del final
        t_amb = Determinar_Ir_Tamb_Ciudad(cboCity.Text,  cboFecha.Text)(1)
' indica el índice del vector.

        If checkAutomatico.Checked = True Then
            Hacer_calculos()
            Posicionar_barras(Ir_sol, p, t_amb)
        End If
    End If

End Sub

Private Sub checkAutomatico_CheckedChanged(sender As Object, e As EventArgs)
Handles checkAutomatico.CheckedChanged
    If checkAutomatico.Checked = True Then
        Hacer_calculos()
        Posicionar_barras(Ir_sol, p, t_amb)
    End If
End Sub

Private Sub btnGraficaBarras_Click(sender As Object, e As EventArgs) Handles
btnGraficaBarras.Click
    Dim x As String
    Dim y(listVarY.SelectedItems.Count - 1) As String
    Dim xmin As Integer
    Dim xmax As Integer

    x = listVarX.SelectedItem

    For i = 0 To (y.Length - 1)
        y(i) = listVarY.SelectedItems(i)
    Next

    xmin = numXmin.Value
    xmax = numXmax.Value

    FormGraficos.chartGrafico.Visible = True
    FormGraficos.chartCompararMateriales.Visible = False
    FormGraficos.chartConsumoAhorro.Visible = False
    FormGraficos.btnHVAC.Visible = False
    FormGraficos.btnTchapa.Visible = False
    FormGraficos.btnTvidrio.Visible = False
    FormGraficos.btnTgap.Visible = False
    FormGraficos.lblBilbao30.Visible = False
    FormGraficos.lblBilbao5.Visible = False
    FormGraficos.lblSevilla5.Visible = False
    FormGraficos.lblSevilla30.Visible = False
    FormGraficos.cboBilbao30.Visible = False
    FormGraficos.cboBilbao5.Visible = False

```

```

FormGraficos.cboSevilla5.Visible = False
FormGraficos.cboSevilla30.Visible = False

Generar_grafico(x, y, xmin, xmax, "column", titulo)

FormGraficos.Show()
End Sub

Private Sub btnGraficaRectas_Click(sender As Object, e As EventArgs) Handles
btnGraficaRectas.Click
  Dim x As String
  Dim y(listVarY.SelectedItems.Count - 1) As String
  Dim xmin As Integer
  Dim xmax As Integer

  If checkSinChapa.Checked = True Then
    p = 0.7
  End If
  x = listVarX.SelectedItem

  For i = 0 To (y.Length - 1)
    y(i) = listVarY.SelectedItems(i)
  Next

  xmin = numXmin.Value
  xmax = numXmax.Value

  FormGraficos.chartGrafico.Visible = True
  FormGraficos.chartCompararMateriales.Visible = False
  FormGraficos.chartConsumoAhorro.Visible = False
  FormGraficos.btnHVAC.Visible = False
  FormGraficos.btnTchapa.Visible = False
  FormGraficos.btnTvidrio.Visible = False
  FormGraficos.btnTgap.Visible = False
  FormGraficos.lblBilbao30.Visible = False
  FormGraficos.lblBilbao5.Visible = False
  FormGraficos.lblSevilla5.Visible = False
  FormGraficos.lblSevilla30.Visible = False
  FormGraficos.cboBilbao30.Visible = False
  FormGraficos.cboBilbao5.Visible = False
  FormGraficos.cboSevilla5.Visible = False
  FormGraficos.cboSevilla30.Visible = False

  Generar_grafico(x, y, xmin, xmax, "spline", titulo)

  FormGraficos.Show()
End Sub

Private Sub btnCompararMateriales_Click(sender As Object, e As EventArgs)
Handles btnCompararMateriales.Click
  listVarY.SelectedItems.Clear()

  FormGraficos.chartGrafico.Visible = False
  FormGraficos.chartConsumoAhorro.Visible = False

```

```

FormGraficos.chartCompararMateriales.Visible = True
FormGraficos.chartCompararMateriales.Titles.Clear()
FormGraficos.chartCompararMateriales.Titles.Add(titulo)
FormGraficos.chartCompararMateriales.Titles(0).Font = New Font("Microsof
Sans Serif", 18, FontStyle.Bold, GraphicsUnit.Pixel)
FormGraficos.chartCompararMateriales.Titles(0).Text = titulo

FormGraficos.Show()
End Sub

Private Sub checkAñadirTitulo_CheckedChanged(sender As Object, e As
EventArgs) Handles checkAñadirTitulo.CheckedChanged
    If checkAñadirTitulo.Checked = True Then
        titulo = InputBox("Introduzca el título del gráfico", "Añadir
título", "")
    End If
End Sub

Private Sub listVarX_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles listVarX.SelectedIndexChanged
    Select Case listVarX.SelectedItem
        Case "t_amb"
            numXmin.Minimum = trackTamb.Minimum
            numXmax.Maximum = trackTamb.Maximum
            numXmin.Value = -10
            numXmax.Value = 30
            numIntervalo.Maximum = trackTamb.Maximum / 10
            numIntervalo.Value = 1
            lblUnidadesXmax.Text = "° C"
            lblUnidadesXmin.Text = lblUnidadesXmax.Text
            lblUnidadesIntervalo.Text = lblUnidadesXmax.Text
            FormGraficos.cboBilbao5.Visible = False
            FormGraficos.cboBilbao30.Visible = False
            FormGraficos.cboSevilla5.Visible = False
            FormGraficos.cboSevilla30.Visible = False
            FormGraficos.lblBilbao5.Visible = False
            FormGraficos.lblBilbao30.Visible = False
            FormGraficos.lblSevilla5.Visible = False
            FormGraficos.lblSevilla30.Visible = False

        Case "Ir_sol"
            numXmin.Minimum = trackIrradiancia.Minimum
            numXmax.Maximum = trackIrradiancia.Maximum
            numXmin.Value = 50
            numXmax.Value = 500
            numIntervalo.Maximum = trackIrradiancia.Maximum / 10
            numIntervalo.Value = 20
            lblUnidadesXmax.Text = "W/m²"
            lblUnidadesXmin.Text = lblUnidadesXmax.Text
            lblUnidadesIntervalo.Text = lblUnidadesXmax.Text
            FormGraficos.cboBilbao5.Visible = False
            FormGraficos.cboBilbao30.Visible = False
            FormGraficos.cboSevilla5.Visible = False
            FormGraficos.cboSevilla30.Visible = False
    
```

```

    FormGraficos.lblBilbao5.Visible = False
    FormGraficos.lblBilbao30.Visible = False
    FormGraficos.lblSevilla5.Visible = False
    FormGraficos.lblSevilla30.Visible = False
  Case "perf_ratio"
    numXmin.Minimum = trackPerfRate.Minimum
    numXmax.Maximum = trackPerfRate.Maximum
    numXmin.Value = 10
    numXmax.Value = 70
    numIntervalo.Maximum = trackPerfRate.Maximum / 10
    numIntervalo.Value = 1
    lblUnidadesXmax.Text = "%"
    lblUnidadesXmin.Text = lblUnidadesXmax.Text
    lblUnidadesIntervalo.Text = lblUnidadesXmax.Text
    FormGraficos.cboBilbao5.Visible = True
    FormGraficos.cboBilbao30.Visible = True
    FormGraficos.cboSevilla5.Visible = True
    FormGraficos.cboSevilla30.Visible = True
    FormGraficos.lblBilbao5.Visible = True
    FormGraficos.lblBilbao30.Visible = True
    FormGraficos.lblSevilla5.Visible = True
    FormGraficos.lblSevilla30.Visible = True
  End Select
End Sub

Private Sub btnConsumoAhorro_Click(sender As Object, e As EventArgs) Handles
btnConsumoAhorro.Click
  Hacer_calculos()
  FormConsumoAhorro.Show()
End Sub

Private Sub checkSinChapa_Click(sender As Object, e As EventArgs) Handles
checkSinChapa.Click
  If checkChapaMaciza.Checked = True Then
    checkChapaMaciza.Checked = False
  End If
End Sub

Private Sub checkChapaMaciza_Click(sender As Object, e As EventArgs) Handles
checkChapaMaciza.Click
  If checkSinChapa.Checked = True Then
    checkSinChapa.Checked = False
  End If
End Sub

Private Sub checkSinChapa_CheckedChanged(sender As Object, e As EventArgs)
Handles checkSinChapa.CheckedChanged
  If checkSinChapa.Checked = True Then
    trackPerfRate.Visible = False
    txtBarraPerfRate.Visible = False
    lblBarraPerfRate.Visible = False
    lblTchapa.Visible = False
    lblTgap.Visible = False
    p = 0
  End If
End Sub

```

```

    pic30cm.Visible = False
    pic5cm.Visible = False
    picMaciza.Visible = False
    picSinChapa.Visible = True

    If checkAutomatico.Checked = True Then
      Hacer_calculos()
    End If
  End If

  If (checkSinChapa.Checked = False) And (checkChapaMaciza.Checked = False)
Then
    trackPerfRate.Visible = True
    txtBarraPerfRate.Visible = True
    lblBarraPerfRate.Visible = True
    p = 0.35
    Hacer_calculos()
  End If

  If checkSinChapa.Checked = False Then
    lblTchapa.Visible = True
    lblTgap.Visible = True

    picSinChapa.Visible = False

    Select Case checkChapaMaciza.Checked
      Case True
        picMaciza.Visible = True
      Case False
        If opt30cm.Checked = True Then
          pic30cm.Visible = True
        ElseIf opt5cm.Checked = True Then
          pic5cm.Visible = True
        End If
    End Select
  End If
End Sub

Private Sub checkChapaMaciza_CheckedChanged(sender As Object, e As EventArgs)
Handles checkChapaMaciza.CheckedChanged
  If checkChapaMaciza.Checked = True Then
    trackPerfRate.Visible = False
    txtBarraPerfRate.Visible = False
    lblBarraPerfRate.Visible = False
    p = 0.9999
    pic30cm.Visible = False
    pic5cm.Visible = False
    picMaciza.Visible = True

    If checkAutomatico.Checked = True Then
      Hacer_calculos()
    End If
  End If
End Sub

```

```

    If (checkChapaMaciza.Checked = False) And (checkSinChapa.Checked = False)
Then
    trackPerfRate.Visible = True
    txtBarraPerfRate.Visible = True
    lblBarraPerfRate.Visible = True
    p = 0.35
    picMaciza.Visible = False

    If opt30cm.Checked = True Then
        pic30cm.Visible = True
    ElseIf opt5cm.Checked = True Then
        pic5cm.Visible = True
    End If
    Hacer_calculos()
End If
End Sub

Private Sub btnGenerarInforme_Click(sender As Object, e As EventArgs) Handles
btnGenerarInforme.Click
    Dim objWord As Word.Application
    Dim objDoc As Word.Document
    Dim nn As Integer
    Dim boSeleccionado As Boolean
    nn = treeProyectos.Nodes.Count()
    boSeleccionado = False

    For i = 0 To (nn - 1)
        If treeProyectos.Nodes(i).IsSelected Then
            boSeleccionado = True
            Asignar_datos_al_proyecto(treeProyectos.Nodes(i).Text)
            Exit For
        End If
    Next

    If boSeleccionado = False Then
        MessageBox.Show("No se ha seleccionado ningún proyecto. Haga click
sobre el " &
            "proyecto al que desee guardar y pulse Guardar.",
"Información",
            MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub
End If

' Iniciar Word y abrir el documento
objWord = CreateObject("Word.Application")
objWord.Visible = True
objDoc = objWord.Documents.Add()

' Insertar el título al comienzo del documento
Dim objPara1 As Word.Paragraph
objPara1 = objDoc.Content.Paragraphs.Add()
objPara1.Range.Font.Bold = True
objPara1.Range.Font.Underline = True
objPara1.Range.Font.Size = 24

```



```

objPara1.Range.Font.Name = "Arial"
objPara1.Range.Text = txtNombreProyecto.Text
objPara1.Format.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphCenter
objPara1.Format.SpaceAfter = 48
objPara1.Range.InsertParagraphAfter()

' Subtítulo: PARÁMETROS DE CÁLCULO
Dim objPara2 As Word.Paragraph
objPara2 = objDoc.Content.Paragraphs.Add()
objPara2.Range.Font.Bold = True
objPara2.Range.Font.Underline = True
objPara2.Range.Font.Size = 13
objPara2.Range.Font.Name = "Arial"
objPara2.Range.Text = "PARÁMETROS DE CÁLCULO"
objPara2.Format.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphLeft
objPara2.Format.SpaceAfter = 24
objPara2.Range.InsertParagraphAfter()

' Sub-subtítulo: Aspectos constructivos de la fachada
Dim objPara3 As Word.Paragraph
objPara3 = objDoc.Content.Paragraphs.Add()
objPara3.Range.Font.Italic = True
objPara3.Range.Font.Underline = True
objPara3.Range.Font.Size = 12
objPara3.Range.Font.Name = "Arial"
objPara3.Range.Text = "Aspectos constructivos de la fachada:"
objPara3.Format.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphLeft
objPara3.Format.SpaceAfter = 24
objPara3.Range.InsertParagraphAfter()

' Añadir tabla con los parámetros de la fachada
Dim objPara4 As Word.Paragraph
Dim objTabFachada As Word.Table
objPara4 = objDoc.Content.Paragraphs.Add()
objTabFachada = objDoc.Content.Tables.Add(objPara4.Range, 5, 4)
objTabFachada.Range.ParagraphFormat.SpaceAfter = 0

For i = 1 To 5
  For j = 1 To 4
    objTabFachada.Cell(i, j).Range.Font.Name = "Arial"
    objTabFachada.Cell(i, j).VerticalAlignment =
Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter
    objTabFachada.Cell(i, j).Range.ParagraphFormat.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphCenter

    If j = 4 Then
      objTabFachada.Cell(i, j).Range.Font.Italic = True
    End If

  Next
Next
Next

```

```

objTabFachada.Cell(1, 1).Merge(objTabFachada.Cell(4, 1))
objTabFachada.Cell(1, 2).Merge(objTabFachada.Cell(3, 2))
objTabFachada.Cell(4, 3).Merge(objTabFachada.Cell(4, 4))
objTabFachada.Cell(5, 3).Merge(objTabFachada.Cell(5, 4))

objTabFachada.Borders.Enable = True

objTabFachada.Cell(1, 1).Range.Text = "CHAPA"
objTabFachada.Cell(1, 1).Range.Font.Bold = True
objTabFachada.Cell(1, 1).Range.Font.Size = 13

objTabFachada.Cell(5, 1).Range.Text = "CÁMARA"
objTabFachada.Cell(5, 1).Range.Font.Bold = True
objTabFachada.Cell(5, 1).Range.Font.Size = 13

objTabFachada.Cell(1, 2).Range.Text = "Material"
objTabFachada.Cell(1, 2).Range.Font.Bold = True
objTabFachada.Cell(1, 2).Range.Font.Size = 12

objTabFachada.Cell(4, 2).Range.Text = "Ratio de perforación"
objTabFachada.Cell(4, 2).Range.Font.Bold = True
objTabFachada.Cell(4, 2).Range.Font.Size = 12

objTabFachada.Cell(5, 2).Range.Text = "Separación"
objTabFachada.Cell(5, 2).Range.Font.Bold = True
objTabFachada.Cell(5, 2).Range.Font.Size = 12

objTabFachada.Cell(1, 3).Range.Text = "Tipo"
objTabFachada.Cell(1, 3).Range.Font.Size = 11

objTabFachada.Cell(2, 3).Range.Text = "Absortividad"
objTabFachada.Cell(2, 3).Range.Font.Size = 11

objTabFachada.Cell(3, 3).Range.Text = "Emisividad"
objTabFachada.Cell(3, 3).Range.Font.Size = 11

objTabFachada.Cell(1, 4).Range.Text = material
objTabFachada.Cell(1, 4).Range.Font.Size = 11

objTabFachada.Cell(2, 4).Range.Text = CStr(100 * alpha_solar_chapa) & "
%"
objTabFachada.Cell(2, 4).Range.Font.Size = 11

objTabFachada.Cell(3, 4).Range.Text = CStr(100 * epsilon_IR_chapa) & " %"
objTabFachada.Cell(3, 4).Range.Font.Size = 11

If (checkSinChapa.Checked = False) And (checkChapaMaciza.Checked = False)
Then
  objTabFachada.Cell(4, 3).Range.Text = CStr(100 * p) & " %"
  objTabFachada.Cell(4, 3).Range.Font.Size = 11
ElseIf checkSinChapa.Checked = True Then
  objTabFachada.Cell(4, 3).Range.Text = "No hay chapa"
ElseIf checkChapaMaciza.Checked = True Then
  
```

```

    objTabFachada.Cell(4, 3).Range.Text = "La chapa es maciza"
  End If

  objTabFachada.Cell(5, 3).Range.Text = separacion & " cm"
  objTabFachada.Cell(5, 3).Range.Font.Size = 11

  objPara4.Format.SpaceAfter = 24
  objPara3.Range.InsertParagraphAfter()

  ' Sub-subtítulo: Ubicación y clima
  Dim objPara5 As Word.Paragraph
  objPara5 = objDoc.Content.Paragraphs.Add()
  objPara5.Range.Font.Italic = True
  objPara5.Range.Font.Underline = True
  objPara5.Range.Font.Size = 12
  objPara5.Range.Font.Name = "Arial"
  objPara5.Range.Text = "Ubicación y clima:"
  objPara5.Format.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphLeft
  objPara5.Format.SpaceAfter = 24
  objPara5.Range.InsertParagraphAfter()

  ' Añadir tabla con la ubicación y el clima
  Dim objPara6 As Word.Paragraph
  Dim objTabUbicacionClima As Word.Table
  objPara6 = objDoc.Content.Paragraphs.Add()
  objTabUbicacionClima = objDoc.Content.Tables.Add(objPara6.Range, 5, 3)
  objTabUbicacionClima.Range.ParagraphFormat.SpaceAfter = 0

  objTabUbicacionClima.Borders.Enable = True

  For i = 1 To 5
    For j = 1 To 3
      objTabUbicacionClima.Cell(i, j).Range.Font.Name = "Arial"
      objTabUbicacionClima.Cell(i, j).VerticalAlignment =
Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter
      objTabUbicacionClima.Cell(i, j).Range.ParagraphFormat.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphCenter

      If j = 3 Then
        objTabUbicacionClima.Cell(i, j).Range.Font.Italic = True
      End If

    Next
  Next

  objTabUbicacionClima.Cell(1, 1).Merge(objTabUbicacionClima.Cell(2, 1))
  objTabUbicacionClima.Cell(3, 1).Merge(objTabUbicacionClima.Cell(5, 1))

  objTabUbicacionClima.Cell(1, 1).Range.Text = "UBICACIÓN"
  objTabUbicacionClima.Cell(1, 1).Range.Font.Bold = True
  objTabUbicacionClima.Cell(1, 1).Range.Font.Size = 13

  objTabUbicacionClima.Cell(3, 1).Range.Text = "CLIMA"

```

```

objTabUbicacionClima.Cell(3, 1).Range.Font.Bold = True
objTabUbicacionClima.Cell(3, 1).Range.Font.Size = 13

objTabUbicacionClima.Cell(1, 2).Range.Text = "Región"
objTabUbicacionClima.Cell(1, 2).Range.Font.Bold = True
objTabUbicacionClima.Cell(1, 2).Range.Font.Size = 12

objTabUbicacionClima.Cell(2, 2).Range.Text = "Ciudad"
objTabUbicacionClima.Cell(2, 2).Range.Font.Bold = True
objTabUbicacionClima.Cell(2, 2).Range.Font.Size = 12

objTabUbicacionClima.Cell(3, 2).Range.Text = "Mes"
objTabUbicacionClima.Cell(3, 2).Range.Font.Bold = True
objTabUbicacionClima.Cell(3, 2).Range.Font.Size = 12

objTabUbicacionClima.Cell(4, 2).Range.Text = "Irradiancia"
objTabUbicacionClima.Cell(4, 2).Range.Font.Bold = True
objTabUbicacionClima.Cell(4, 2).Range.Font.Size = 12

objTabUbicacionClima.Cell(5, 2).Range.Text = "Temperatura ambiente"
objTabUbicacionClima.Cell(5, 2).Range.Font.Bold = True
objTabUbicacionClima.Cell(5, 2).Range.Font.Size = 12

For i = 1 To 5
    objTabUbicacionClima.Cell(i, 3).Range.Font.Size = 11
Next

If cboRegión.Text = "Seleccione una región" Then
    objTabUbicacionClima.Cell(1, 3).Range.Text = "No se ha seleccionado
ninguna región"
Else
    objTabUbicacionClima.Cell(1, 3).Range.Text = cboRegión.Text
End If

If cboCity.Text = "Seleccione una ciudad" Then
    objTabUbicacionClima.Cell(2, 3).Range.Text = "No se ha seleccionado
ninguna ciudad"
Else
    objTabUbicacionClima.Cell(2, 3).Range.Text = cboCity.Text
End If

If cboFecha.Text = "Seleccione un mes" Then
    objTabUbicacionClima.Cell(3, 3).Range.Text = "No se ha seleccionado
ningún mes"
Else
    objTabUbicacionClima.Cell(3, 3).Range.Text = cboFecha.Text
End If

objTabUbicacionClima.Cell(4, 3).Range.Text = CStr(Ir_sol) & " W/m²"
objTabUbicacionClima.Cell(5, 3).Range.Text = CStr(Round(t_amb - 273, 2))
& " °C"

objPara6.Format.SpaceAfter = 48
objPara6.Range.InsertParagraphAfter()

```

```

' Subtítulo: RESULTADOS
Dim objPara7 As Word.Paragraph
objPara7 = objDoc.Content.Paragraphs.Add()
objPara7.Range.Font.Bold = True
objPara7.Range.Font.Underline = True
objPara7.Range.Font.Size = 13
objPara7.Range.Font.Name = "Arial"
objPara7.Range.Text = "RESULTADOS"
objPara7.Format.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphLeft
objPara7.Format.SpaceAfter = 24
objPara7.Range.InsertParagraphAfter()

' Sub-subtítulo: Temperaturas en la fachada y HVAC
Dim objPara8 As Word.Paragraph
objPara8 = objDoc.Content.Paragraphs.Add()
objPara8.Range.Font.Italic = True
objPara8.Range.Font.Underline = True
objPara8.Range.Font.Size = 12
objPara8.Range.Font.Name = "Arial"
objPara8.Range.Text = "Temperaturas en la fachada y HVAC:"
objPara8.Format.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphLeft
objPara8.Format.SpaceAfter = 24
objPara8.Range.InsertParagraphAfter()

' Añadir tabla de resultados de temperaturas y HVAC
Dim objPara9 As Word.Paragraph
Dim objTabResultados As Word.Table
objPara9 = objDoc.Content.Paragraphs.Add()
objTabResultados = objDoc.Content.Tables.Add(objPara9.Range, 2, 4)
objTabResultados.Range.ParagraphFormat.SpaceAfter = 0

objTabResultados.Borders.Enable = True

Dim strVarRes(3) As String
strVarRes = {"T_chapa", "T_vidrio", "T_cámara", "HVAC"}

For i = 1 To 2
  For j = 1 To 4
    objTabResultados.Cell(i, j).Range.Font.Name = "Arial"
    objTabResultados.Cell(i, j).VerticalAlignment =
Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter
    objTabResultados.Cell(i, j).Range.ParagraphFormat.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphCenter

    If i = 1 Then
      objTabResultados.Cell(1, j).Range.Text = strVarRes(j - 1)
      objTabResultados.Cell(1, j).Range.Font.Bold = True
      objTabResultados.Cell(1, j).Range.Font.Size = 12
    ElseIf i = 2 Then
      objTabResultados.Cell(2, j).Range.Font.Bold = False
      objTabResultados.Cell(2, j).Range.Font.Italic = True
  
```

```

        objTabResultados.Cell(2, j).Range.Font.Size = 11
    End If

    Next
Next

If checkSinChapa.Checked = False Then
    objTabResultados.Cell(2, 1).Range.Text =
Microsoft.VisualBasic.Right(lblTchapa.Text, 10)
    objTabResultados.Cell(2, 3).Range.Text =
Microsoft.VisualBasic.Right(lblTgap.Text, 10)
Else
    objTabResultados.Cell(2, 1).Range.Font.Italic = True
    objTabResultados.Cell(2, 1).Range.Text = "No hay chapa"
    objTabResultados.Cell(2, 3).Range.Font.Italic = True
    objTabResultados.Cell(2, 3).Range.Text = "No hay cámara"
End If

objTabResultados.Cell(2, 2).Range.Text =
Microsoft.VisualBasic.Right(lblTvidrio.Text, 10)
objTabResultados.Cell(2, 4).Range.Text = lblHVAC.Text & " W/m²"

objPara9.Format.SpaceAfter = 96
objPara9.Range.InsertParagraphAfter()

' Añadir tablas de resultados de consumo y ahorro
If (cboCity.Text = "Bilbao") Or (cboCity.Text = "Sevilla") Then
    Dim objPara10 As Word.Paragraph
    objPara10 = objDoc.Content.Paragraphs.Add()
    objPara10.Range.Font.Italic = True
    objPara10.Range.Font.Underline = True
    objPara10.Range.Font.Size = 12
    objPara10.Range.Font.Name = "Arial"
    objPara10.Range.Text = "Consumo y ahorro anual:"
    objPara10.Format.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphLeft
    objPara10.Format.SpaceAfter = 24
    objPara10.Range.InsertParagraphAfter()

    Dim cargas_y_ahorro(5) As String
    Dim sin_chapa_bilbao(2) As Double
    Dim sin_chapa_sevilla(2) As Double
    cargas_y_ahorro = Calcular_Ilu_Heat_AC(cboCity.Text, material, p,
separacion)
    sin_chapa_bilbao = {845.6753, 1832.949, 4088.073}
    sin_chapa_sevilla = {813.4565, 97.28376, 11780.18}

' Tabla de consumo y ahorro
Dim objParaCons As Word.Paragraph
Dim objTabCons As Word.Table
objParaCons = objDoc.Content.Paragraphs.Add()
objTabCons = objDoc.Content.Tables.Add(objParaCons.Range, 3, 5)
objTabCons.Range.ParagraphFormat.SpaceAfter = 0
objTabCons.Borders.Enable = True

```

```

Dim strVarConsIluHeat(3) As String
Dim strConsAhorro(1) As String
strVarConsIluHeat = {"Iluminación", "Calefacción", "Aire
acondicionado", "TOTAL"}
strConsAhorro = {"Consumo", "Ahorro"}

For i = 1 To 3
  For j = 1 To 5
    objTabCons.Cell(i, j).Range.Font.Name = "Arial"
    objTabCons.Cell(i, j).VerticalAlignment =
Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter
    objTabCons.Cell(i, j).Range.ParagraphFormat.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphCenter

    If (i = 1) And (j > 1) Then
      objTabCons.Cell(i, j).Range.Font.Bold = True
      objTabCons.Cell(i, j).Range.Font.Size = 12
      objTabCons.Cell(i, j).Range.Text = strVarConsIluHeat(j -
2)
    ElseIf (i > 1) And (j = 1) Then
      objTabCons.Cell(i, j).Range.Font.Bold = True
      objTabCons.Cell(i, j).Range.Font.Size = 12
      objTabCons.Cell(i, j).Range.Text = strConsAhorro(i - 2)
    Else
      objTabCons.Cell(i, j).Range.Font.Italic = True
      objTabCons.Cell(i, j).Range.Font.Size = 11

      If (i = 2) And (j > 1) And (j < 5) And
(checkSinChapa.Checked = False) Then
        objTabCons.Cell(i, j).Range.Text = cargas_y_ahorro(j
- 2) & " kWh"
      ElseIf (i = 3) And (j > 1) And (j < 5) And
(checkSinChapa.Checked = False) Then
        objTabCons.Cell(i, j).Range.Text = cargas_y_ahorro(j
- 2 + 3) & " %"
      End If
    End If
  Next
Next

If (checkSinChapa.Checked = False) And (cboCity.Text = "Bilbao") Then
  objTabCons.Cell(2, 5).Range.Text = CStr(CDbl(cargas_y_ahorro(0))
+ CDbl(cargas_y_ahorro(1)) + CDbl(cargas_y_ahorro(2))) & " kWh"
  objTabCons.Cell(3, 5).Range.Text = CStr(Round((-
CDbl(cargas_y_ahorro(0)) - CDbl(cargas_y_ahorro(1)) - CDbl(cargas_y_ahorro(2)) +
sin_chapa_bilbao(0) + sin_chapa_bilbao(1) + sin_chapa_bilbao(2)) /
(sin_chapa_bilbao(0) + sin_chapa_bilbao(1) + sin_chapa_bilbao(2)) * 100, 2)) & "
%"
  ElseIf (checkSinChapa.Checked = True) And (cboCity.Text = "Bilbao")
Then
    objTabCons.Cell(2, 2).Range.Text = CStr(sin_chapa_bilbao(0)) & "
kWh"
  
```

```

    objTabCons.Cell(2, 3).Range.Text = CStr(sin_chapa_bilbao(1)) & "
kWh"
    objTabCons.Cell(2, 4).Range.Text = CStr(sin_chapa_bilbao(2)) & "
kWh"
    objTabCons.Cell(2, 5).Range.Text = CStr(sin_chapa_bilbao(0) +
sin_chapa_bilbao(1) + sin_chapa_bilbao(2)) & " kWh"

    For j = 2 To 5
      objTabCons.Cell(3, j).Range.Text = "0 %"
    Next
  End If

  If (checkSinChapa.Checked = False) And (cboCity.Text = "Sevilla")
  Then
    objTabCons.Cell(2, 5).Range.Text = CStr(CDb1(cargas_y_ahorro(0))
+ CDb1(cargas_y_ahorro(1)) + CDb1(cargas_y_ahorro(2))) & " kWh"
    objTabCons.Cell(3, 5).Range.Text =
CStr(Round((sin_chapa_sevilla(0) + sin_chapa_sevilla(1) + sin_chapa_sevilla(2)) -
(CDb1(cargas_y_ahorro(0)) + CDb1(cargas_y_ahorro(1)) + CDb1(cargas_y_ahorro(2)))
/ (sin_chapa_sevilla(0) + sin_chapa_sevilla(1) + sin_chapa_sevilla(2)), 2)) & "
%"
    ElseIf (checkSinChapa.Checked = True) And (cboCity.Text = "Sevilla")
  Then
    objTabCons.Cell(2, 2).Range.Text = CStr(sin_chapa_sevilla(0)) & "
kWh"
    objTabCons.Cell(2, 3).Range.Text = CStr(sin_chapa_sevilla(1)) & "
kWh"
    objTabCons.Cell(2, 4).Range.Text = CStr(sin_chapa_sevilla(2)) & "
kWh"
    objTabCons.Cell(2, 5).Range.Text = CStr(sin_chapa_sevilla(0) +
sin_chapa_sevilla(1) + sin_chapa_sevilla(2)) & " kWh"

    For j = 2 To 5
      objTabCons.Cell(3, j).Range.Text = "0 %"
    Next
  End If

  End If

  End Sub

  Private Sub btnNuevoProyecto_Click(sender As Object, e As EventArgs) Handles
btnNuevoProyecto.Click
    Dim nombre_proyecto As String

    nombre_proyecto = InputBox("Introduzca el nombre del proyecto", "Nuevo
proyecto", "")
    Generar_proyecto(nombre_proyecto)
    Rellenar_proyecto(nombre_proyecto)
    txtNombreProyecto.Text = nombre_proyecto

  End Sub

```



```

Private Sub treeProyectos_NodeMouseDoubleClick(sender As Object, e As
EventArgs) Handles treeProyectos.NodeMouseDoubleClick
    Dim nn As Integer
    nn = treeProyectos.Nodes.Count()

    For i = 0 To (nn - 1)
        If treeProyectos.Nodes(i).IsSelected Then
            txtNombreProyecto.Text = treeProyectos.Nodes(i).Text
            Asignar_datos_al_proyecto(treeProyectos.Nodes(i).Text)
            Exit For
        End If
    Next
End Sub

Private Sub treeProyectos_AfterSelect(sender As Object, e As EventArgs)
Handles treeProyectos.AfterSelect
    Dim nn As Integer
    nn = treeProyectos.Nodes.Count()

    For i = 0 To (nn - 1)
        If treeProyectos.Nodes(i).IsSelected Then
            txtNombreProyecto.Text = treeProyectos.Nodes(i).Text
            Exit For
        End If
    Next
End Sub

Private Sub btnCambiarNombre_Click(sender As Object, e As EventArgs) Handles
btnCambiarNombre.Click
    Dim nuevo_nombre As String
    Dim nn As Integer
    Dim boNombreCambiado As Boolean = False
    nn = treeProyectos.Nodes.Count()

    For i = 0 To (nn - 1)
        If treeProyectos.Nodes(i).IsSelected Then
            nuevo_nombre = InputBox("Introduzca el nombre del proyectos",
                "Cambiar nombre", "")

            If Not (nuevo_nombre = "") Then
                treeProyectos.Nodes(i).Text = nuevo_nombre
                txtNombreProyecto.Text = nuevo_nombre
                boNombreCambiado = True
            Else
                MessageBox.Show("No se ha introducido ningún nombre para el
proyecto " &
                    treeProyectos.Nodes(i).Text, "Información",
                    MessageBoxButtons.OK,
                    MessageBoxIcon.Information)
                boNombreCambiado = True
            End If
            Exit For
        End If
    Next

```

```

        If boNombreCambiado = False Then
            MsgBox.Show("No se ha seleccionado ningún proyecto. Haga click
sobre el " &
                "proyecto al que desee cambiar el nombre y pulse
Renombrar.", "Información",
                MsgBoxButtons.OK, MsgBoxIcon.Information)
        End If

    End Sub

Private Sub btnGuardarProyecto_Click(sender As Object, e As EventArgs)
Handles btnGuardarProyecto.Click
    Dim nn As Integer
    Dim boSeleccionado As Boolean
    nn = treeProyectos.Nodes.Count()
    boSeleccionado = False

    For i = 0 To (nn - 1)
        If treeProyectos.Nodes(i).IsSelected Then
            boSeleccionado = True
            Guardar_proyecto()
            Exit For
        End If
    Next

    If boSeleccionado = False Then
        MsgBox.Show("No se ha seleccionado ningún proyecto. Haga click
sobre el " &
            "proyecto al que desee guardar y pulse Guardar.",
"Información",
            MsgBoxButtons.OK, MsgBoxIcon.Information)
    End If

End Sub

Private Sub btnAbrirProyecto_Click(sender As Object, e As EventArgs) Handles
btnAbrirProyecto.Click
    Abrir_proyecto()
End Sub

Private Sub btnActualizarProyecto_Click(sender As Object, e As EventArgs)
Handles btnActualizarProyecto.Click
    Dim nn As Integer
    Dim boSeleccionado As Boolean
    nn = treeProyectos.Nodes.Count()
    boSeleccionado = False

    For i = 0 To (nn - 1)
        If treeProyectos.Nodes(i).IsSelected Then
            boSeleccionado = True
            Rellenar_proyecto(treeProyectos.Nodes(i).Text)
            Exit For
        End If
    Next

```

```

        If boSeleccionado = False Then
            MsgBox.Show("No se ha seleccionado ningún proyecto. Haga click
sobre el " &
                "proyecto al que desee actualizar y pulse
Actualizar.", "Información",
                    MsgBoxButtons.OK, MessageBoxIcon.Information)
        End If
    End Sub

    Private Sub btnEliminarProyecto_Click(sender As Object, e As EventArgs)
Handles btnEliminarProyecto.Click
        Dim nn As Integer
        Dim boSeleccionado As Boolean
        nn = treeProyectos.Nodes.Count()
        boSeleccionado = False

        For i = 0 To (nn - 1)
            If treeProyectos.Nodes(i).IsSelected Then
                boSeleccionado = True
                treeProyectos.Nodes(i).Remove()
            Exit For
            End If
        Next

        If boSeleccionado = False Then
            MsgBox.Show("No se ha seleccionado ningún proyecto. Haga click
sobre el " &
                "nombre del proyecto que desee eliminar y pulse
Eliminar.", "Información",
                    MsgBoxButtons.OK, MessageBoxIcon.Information)
            Exit Sub
        End If

        Try
            For i = 0 To (nn - 1)
                If treeProyectos.Nodes(i).IsSelected Then
                    txtNombreProyecto.Text = treeProyectos.SelectedNode.Text
                Exit For
            End If
        Next
        Catch
            Exit Sub
        End Try

    End Sub

End Class

```

FormGraficos

```
Imports System.IO
```

```
Imports System.Drawing.Imaging
```

```
Public Class FormGraficos
```

```
    Dim num_graf_1 As Integer = 0
```

```
    Dim num_graf_2 As Integer = 0
```

```
    Private Sub FormGraficos_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
        Me.Icon = FormPrincipal.Icon
```

```
        btnTchapa.BringToFront()
```

```
        btnTvidrio.BringToFront()
```

```
        btnTgap.BringToFront()
```

```
        btnHVAC.BringToFront()
```

```
        If Not (FormPrincipal.listVarX.SelectedItem = "perf_ratio") Then
```

```
            lblBilbao5.Visible = False
```

```
            lblBilbao30.Visible = False
```

```
            lblSevilla5.Visible = False
```

```
            lblSevilla30.Visible = False
```

```
            cboBilbao5.Visible = False
```

```
            cboBilbao30.Visible = False
```

```
            cboSevilla5.Visible = False
```

```
            cboSevilla30.Visible = False
```

```
        End If
```

```
    End Sub
```

```
    Private Sub btnTchapa_Click(sender As Object, e As EventArgs) Handles btnTchapa.Click
```

```
        Dim vble_x As String = FormPrincipal.listVarX.SelectedItem
```

```
        Dim xmax As Integer = FormPrincipal.numXmax.Value
```

```
        Dim xmin As Integer = FormPrincipal.numXmin.Value
```

```
        Dim titulo_grafico As String = FormPrincipal.titulo
```

```
        Grafico_materiales(vble_x, "t_chapa", titulo_grafico, xmax, xmin)
```

```
    End Sub
```

```
    Private Sub btnTvidrio_Click(sender As Object, e As EventArgs) Handles btnTvidrio.Click
```

```
        Dim vble_x As String = FormPrincipal.listVarX.SelectedItem
```

```
        Dim xmax As Integer = FormPrincipal.numXmax.Value
```

```
        Dim xmin As Integer = FormPrincipal.numXmin.Value
```

```
        Dim titulo_grafico As String = FormPrincipal.titulo
```

```
        Grafico_materiales(vble_x, "t_vidrio", titulo_grafico, xmax, xmin)
```

```
    End Sub
```

```
    Private Sub btnTgap_Click(sender As Object, e As EventArgs) Handles btnTgap.Click
```

```

Dim vble_x As String = FormPrincipal.listVarX.SelectedItem
Dim xmax As Integer = FormPrincipal.numXmax.Value
Dim xmin As Integer = FormPrincipal.numXmin.Value
Dim titulo_grafico As String = FormPrincipal.titulo

Grafico_materiales(vble_x, "t_gap", titulo_grafico, xmax, xmin)
End Sub

Private Sub btnHVAC_Click(sender As Object, e As EventArgs) Handles
btnHVAC.Click
Dim vble_x As String = FormPrincipal.listVarX.SelectedItem
Dim xmax As Integer = FormPrincipal.numXmax.Value
Dim xmin As Integer = FormPrincipal.numXmin.Value
Dim titulo_grafico As String = FormPrincipal.titulo

Grafico_materiales(vble_x, "HVAC", titulo_grafico, xmax, xmin)
End Sub

Private Sub cboBilbao5_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles cboBilbao5.SelectedIndexChanged
Dim xmax As Integer = FormPrincipal.numXmax.Value
Dim xmin As Integer = FormPrincipal.numXmin.Value
Dim titulo_grafico As String = FormPrincipal.titulo

Select Case cboBilbao5.SelectedItem
Case "Iluminación"
Grafico_materiales("perf_ratio", "Bilbao5_Ilu", titulo_grafico,
xmax, xmin)
chartConsumoAhorro.Titles(1).Text = "Ahorro (%)"
Case "Calefacción"
Grafico_materiales("perf_ratio", "Bilbao5_Heat", titulo_grafico,
xmax, xmin)
chartConsumoAhorro.Titles(1).Text = "Incremento (%)"
Case "Aire acondicionado"
Grafico_materiales("perf_ratio", "Bilbao5_AC", titulo_grafico,
xmax, xmin)
chartConsumoAhorro.Titles(1).Text = "Ahorro (%)"
End Select
End Sub

Private Sub cboSevilla5_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles cboSevilla5.SelectedIndexChanged
Dim xmax As Integer = FormPrincipal.numXmax.Value
Dim xmin As Integer = FormPrincipal.numXmin.Value
Dim titulo_grafico As String = FormPrincipal.titulo

Select Case cboSevilla5.SelectedItem
Case "Iluminación"
Grafico_materiales("perf_ratio", "Sevilla5_Ilu", titulo_grafico,
xmax, xmin)
chartConsumoAhorro.Titles(1).Text = "Ahorro (%)"
Case "Calefacción"
Grafico_materiales("perf_ratio", "Sevilla5_Heat", titulo_grafico,
xmax, xmin)

```

```

        chartConsumoAhorro.Titles(1).Text = "Incremento (%)"
    Case "Aire acondicionado"
        Grafico_materiales("perf_ratio", "Sevilla5_AC", titulo_grafico,
xmax, xmin)
        chartConsumoAhorro.Titles(1).Text = "Ahorro (%)"
    End Select
End Sub

Private Sub cboBilbao30_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles cboBilbao30.SelectedIndexChanged
    Dim xmax As Integer = FormPrincipal.numXmax.Value
    Dim xmin As Integer = FormPrincipal.numXmin.Value
    Dim titulo_grafico As String = FormPrincipal.titulo

    Select Case cboBilbao30.SelectedItem
        Case "Iluminación"
            Grafico_materiales("perf_ratio", "Bilbao30_Ilu", titulo_grafico,
xmax, xmin)
            chartConsumoAhorro.Titles(1).Text = "Ahorro (%)"
        Case "Calefacción"
            Grafico_materiales("perf_ratio", "Bilbao30_Heat", titulo_grafico,
xmax, xmin)
            chartConsumoAhorro.Titles(1).Text = "Incremento (%)"
        Case "Aire acondicionado"
            Grafico_materiales("perf_ratio", "Bilbao30_AC", titulo_grafico,
xmax, xmin)
            chartConsumoAhorro.Titles(1).Text = "Ahorro (%)"
    End Select
End Sub

Private Sub cboSevilla30_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles cboSevilla30.SelectedIndexChanged
    Dim xmax As Integer = FormPrincipal.numXmax.Value
    Dim xmin As Integer = FormPrincipal.numXmin.Value
    Dim titulo_grafico As String = FormPrincipal.titulo

    Select Case cboSevilla30.SelectedItem
        Case "Iluminación"
            Grafico_materiales("perf_ratio", "Sevilla30_Ilu", titulo_grafico,
xmax, xmin)
            chartConsumoAhorro.Titles(1).Text = "Ahorro (%)"
        Case "Calefacción"
            Grafico_materiales("perf_ratio", "Sevilla30_Heat",
titulo_grafico, xmax, xmin)
            chartConsumoAhorro.Titles(1).Text = "Incremento (%)"
        Case "Aire acondicionado"
            Grafico_materiales("perf_ratio", "Sevilla30_AC", titulo_grafico,
xmax, xmin)
            chartConsumoAhorro.Titles(1).Text = "Ahorro (%)"
    End Select
End Sub

Private Sub btnGuardarGrafico_Click(sender As Object, e As EventArgs) Handles
btnGuardarGrafico.Click

```

```

Try
  Dim path As String

  If sfdGuardarGrafico.ShowDialog() <> DialogResult.Cancel Then
    path = sfdGuardarGrafico.FileName
    If chartGrafico.Visible = True Then
      If Not File.Exists(path & "\" & chartGrafico.Titles(0).Text &
".png") Then
        chartGrafico.SaveImage(path & "\" &
chartGrafico.Titles(0).Text & ".png",
          ImageFormat.Png)
      Else
        num_graf_1 = num_graf_1 + 1
        chartGrafico.SaveImage(path & "\" &
chartGrafico.Titles(0).Text & "(" & CStr(num_graf_1) &
          ") .png", ImageFormat.Png)
      End If

      ElseIf chartConsumoAhorro.Visible = True Then
        If Not File.Exists(path & "\" &
chartConsumoAhorro.Titles(0).Text & ".png") Then
          chartConsumoAhorro.SaveImage(path & "\" &
chartConsumoAhorro.Titles(0).Text & ".png",
            ImageFormat.Png)
        Else
          num_graf_2 = num_graf_2 + 1
          chartConsumoAhorro.SaveImage(path & "\" &
chartConsumoAhorro.Titles(0).Text & "(" & CStr(num_graf_2) &
            ") .png", ImageFormat.Png)
        End If

        ElseIf chartCompararMateriales.Visible = True Then
          If Not File.Exists(path & "\" &
chartCompararMateriales.Titles(0).Text & ".png") Then
            chartCompararMateriales.SaveImage(path & "\" &
chartCompararMateriales.Titles(0).Text & ".png",
              ImageFormat.Png)
          Else
            num_graf_1 = num_graf_1 + 1
            chartCompararMateriales.SaveImage(path & "\" &
chartCompararMateriales.Titles(0).Text & "(" & CStr(num_graf_1) &
              ") .png", ImageFormat.Png)
          End If

          End If
        End If

      Catch
        MessageBox.Show("Esta aplicación no posee los permisos necesarios
para realizar" &
          "esta acción. Ejecútela como administrador para poder
guardar " &
          "el gráfico.", "Aviso", MessageBoxButtons.OKCancel,
MessageBoxIcon.Exclamation)
    
```

```

    End Try
  End Sub
End Class

```

FormConsumoAhorro

```

Public Class FormConsumoAhorro
  Private Sub FormConsumoAhorro_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
    Me.Icon = FormPrincipal.Icon
  End Sub
End Class

```

FormUserDefMaterial

```

Public Class FormUserDefMaterial
  Private Sub FormUserDefMaterial_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
    Me.Icon = FormPrincipal.Icon
  End Sub

  Private Sub btnAceptar_Click(sender As Object, e As EventArgs) Handles
btnAceptar.Click
    If IsNumeric(txtAlpha.Text) And IsNumeric(txtEpsilon.Text) Then

      alpha_solar_chapa = Cdbl(txtAlpha.Text) / 100
      epsilon_IR_chapa = Cdbl(txtEpsilon.Text) / 100

      If FormPrincipal.checkAutomatico.Checked = True Then
        Hacer_calculos()
      End If

      Me.Close()

    ElseIf MessageBox.Show("Los valores de las propiedades no son válidos,
¿desea modificarlos?",
                           "Información", MessageBoxButtons.YesNo,
MessageBoxIcon.Information) =
                           Windows.Forms.DialogResult.No Then

      txtAlpha.Text = ""
      txtEpsilon.Text = ""
      FormPrincipal.optUserDefined.Checked = False
      FormPrincipal.optAlumnAnod.Checked = True

      Me.Close()

    End If

  End Sub
End Class

```



```

Private Sub btnCancelar_Click(sender As Object, e As EventArgs) Handles
btnCancelar.Click
    txtAlpha.Text = ""
    txtEpsilon.Text = ""
    FormPrincipal.optUserDefined.Checked = False
    FormPrincipal.optAlumnAnod.Checked = True

    Me.Close()
End Sub
End Class
  
```

FormIrradTamb

```

Public Class FormIrradTamb
    Private Sub btnAceptar2_Click(sender As Object, e As EventArgs) Handles
btnAceptar2.Click

        Ir_sol = Cdbl(numIrradiancia.Value)
        t_amb = Cdbl(numTamb.Value) + 273

        FormPrincipal.cboRegión.Text = "Seleccione una región"
        FormPrincipal.cboCity.Text = "Seleccione una ciudad"
        FormPrincipal.cboFecha.Text = "Seleccione un mes"
        FormPrincipal.cboTipoDia.Text = "Seleccione el tipo de día"

        If FormPrincipal.checkAutomatico.Checked = True Then
            Hacer_calculos()
            Posicionar_barras(Ir_sol, p, t_amb)
        End If

        Me.Close()

    End Sub

    Private Sub btnCancelar2_Click(sender As Object, e As EventArgs) Handles
btnCancelar2.Click
        numIrradiancia.Value = Ir_sol
        numTamb.Value = t_amb - 273

        Me.Close()

    End Sub
End Class
  
```

modCalculos

```
Imports System.Math
```

```
Module modCalculos
```

```
    ' Declaración de constantes globales
```

```
    Public Const sigma As Double = 0.000000568 ' Constante de Boltzmann
```

```
    Public Const F_gnd As Double = 0.5 ' Factor de forma respecto del suelo
```

```
    Public Const F_sky As Double = 0.5 * (0.5 ^ (1 / 2)) ' Factor de forma  
respecto del cielo
```

```
    Public Const F_air As Double = 0.5 * (1 - 0.5 ^ (1 / 2)) ' Factor de forma  
respecto al aire
```

```
    Public Const F_int_chapa As Double = 0.18
```

```
    Public Const F_int_vidrio As Double = 0.18
```

```
    Public Const F_chapa_int As Double = 0.74
```

```
    Public Const F_vidrio_int As Double = 0.74
```

```
    Public Const epsilon_IR_interior = 0.9
```

```
    Public Const factor_angulo = 0.7
```

```
    ' Declaración de variables globales
```

```
    Public alpha_solar_chapa As Double
```

```
    Public alpha_solar_vidrio As Double = 0.03 ' ¿Por qué en MATLAB usa  
alpha_solar_vidrio = 0.1?
```

```
    Public epsilon_IR_chapa As Double
```

```
    Public epsilon_IR_vidrio As Double = 0.9 ' Glass(Dbl. Clr. 6mm/13mm air)
```

```
    Public strCityOption As String
```

```
    Public Ir_sol As Double ' Irradiancia solar en W/m²
```

```
    Public t_amb As Double ' Temperatura ambiente en K
```

```
    Public h_camara As Double = 12
```

```
    Public p As Double
```

```
    Public t_int As Double = 294
```

```
    Public material As String
```

```
    Public separacion As Double
```

```
    Public ciudad As String
```

```
    Public tr_lw_vidrio As Double = 0.2
```

```
    Public tr_sw_vidrio As Double = 0.8
```

```
Private Function Linsolve_LU(ByVal A As Double(,),
```

```
    ByVal b As Double(,)) As Double(,)
```

```
    ' Esta función resuelve un sistema lineal de la forma
```

```
    '     A*x = b
```

```
    ' mediante factorización LU.
```

```
    ' A es una matriz de 2x2 y b es un vector columna 2x1
```

```
    ' x tiene 3 posiciones --> ver comentario de Calcular_tchapa_tvidrio
```

```
    ' Declaración de variables locales de la función
```

```
    Dim x(2, 0) As Double
```

```
    Dim L(1, 1) As Double
```

```
    Dim U(1, 1) As Double
```

```
    Dim y(1, 0) As Double
```

```
    ' Operaciones
```

```
    L = {{1, 0}, {(A(1, 0) / A(0, 0)), 1}}
```

```

    U = {{A(0, 0), A(0, 1)}, {0, (A(1, 1) - A(0, 1) * A(1, 0) / A(0, 0))}}
    y = {{b(0, 0) / L(0, 0)}, {(b(1, 0) - L(1, 0) * b(0, 0) / L(0, 0)) / L(1,
1)}}
    x = {{(y(0, 0) - U(0, 1) / U(1, 1) * y(1, 0)) / U(0, 0)}, {y(1, 0) / U(1,
1)}}, {320}}
  
```

```

    Return x
  End Function
  
```

```

Private Function Signo(x As Double) As Double
  Dim value As Double
  
```

```

  If x > 0 Then
    value = 1
  ElseIf x < 0 Then
    value = -1
  ElseIf x = 0 Then
    value = 0
  End If
  
```

```

  Return value
End Function
  
```

```

Public Function Calcular_tchapa_tvidrio(ByVal x0 As Double(,),
    ByVal epsilon_IR_chapa As Double,
    ByVal epsilon_IR_vidrio As Double,
    ByVal h_camara As Double,
    ByVal p As Double,
    ByVal t_amb As Double,
    ByVal t_int As Double,
    ByVal alpha_solar_chapa As Double,
    ByVal alpha_solar_vidrio As Double,
    ByVal Ir_sol As Double) As Double(,)
  
```

```

' Esta función aplica el algoritmo de Newton-Raphson
' para resolver el sistema no lineal de ecuaciones
' que define las temperaturas de la chapa y del vidrio.
'
' Para ello, utiliza un vector I2 que representa las ecuaciones
' del sistema, y una matriz J que es el Jacobiano de dicho
' sistema de ecuaciones. Ambas dos han sido calculadas analíticamente.
'
' El vector x tiene tres columnas. Para el algoritmo de resolución se
utilizan
' solamente 2. La tercera posición devuelve la t_gap tras haber realizado
todas las iteraciones.
  
```

```

' Declaración de variables locales de la función
Dim x(2, 0) As Double
Dim t_chapa As Double
Dim t_vidrio As Double
Dim I2(1, 0) As Double
Dim J(1, 1) As Double
Dim x1(2, 0) As Double
  
```

```

Dim errabs_chapa As Double
Dim errabs_vidrio As Double
Dim iter As Integer

' Operaciones
errabs_chapa = 1
errabs_vidrio = 1
iter = 1

Do While ((Abs(errabs_chapa) > 0.00001) Or (Abs(errabs_vidrio) >
0.00001)) And (iter < 1000)
  t_chapa = x0(0, 0)
  t_vidrio = x0(1, 0)

  ' Definir el vector columna I2
  I2(0, 0) = ((0.000000568 * p - 0.000000568) * (t_chapa ^ 4 - 1.0 *
t_vidrio ^ 4)) / (1 / epsilon_IR_chapa + 1 / epsilon_IR_vidrio - 1.0) - 1.0 *
Ir_sol * alpha_solar_chapa * (p - 1.0) - 0.000000568 * epsilon_IR_chapa * (p -
1.0) * (t_amb ^ 4 - 1.0 * t_chapa ^ 4) - 1.0 * (p - 1.0) * ((1.31 * Abs(t_amb -
1.0 * t_chapa) ^ (1 / 3) + 9.0) * (t_amb - 1.0 * t_chapa) + 1.52 * Abs(t_chapa -
1.0 * t_vidrio) ^ (1 / 3) * (0.5 * t_amb - 0.75 * t_chapa + 0.25 * t_vidrio)) -
0.06912 * Ir_sol * alpha_solar_chapa * p * (p - 1.0)

  I2(1, 0) = 0.00000042032 * epsilon_IR_vidrio * (t_int ^ 4 - 1.0 *
t_vidrio ^ 4) + 1.52 * Abs(t_chapa - 1.0 * t_vidrio) ^ (1 / 3) * (0.5 * t_amb +
0.25 * t_chapa - 0.75 * t_vidrio) + 1.31 * Abs(t_int - 1.0 * t_vidrio) ^ (1 / 3)
* (t_int - 1.0 * t_vidrio) - 0.038016 * Ir_sol * p * (p - 1.0) + 0.7 * Ir_sol *
alpha_solar_vidrio * p - (1.0 * (0.000000568 * p - 0.000000568) * (t_chapa ^ 4
- 1.0 * t_vidrio ^ 4)) / (1 / epsilon_IR_chapa + 1 / epsilon_IR_vidrio - 1.0) +
0.000000568 * epsilon_IR_vidrio * p * (t_amb ^ 4 - 1.0 * t_vidrio ^ 4)

  ' Definir la matriz Jacobiana
  J(0, 0) = (p - 1.0) * (1.31 * Abs(t_amb - 1.0 * t_chapa) ^ (1 / 3) +
1.14 * Abs(t_chapa - 1.0 * t_vidrio) ^ (1 / 3) + (0.50666666666666671 * Signo(1.0
* t_vidrio - t_chapa) * (0.5 * t_amb - 0.75 * t_chapa + 0.25 * t_vidrio)) /
Abs(t_chapa - 1.0 * t_vidrio) ^ (2 / 3) - (0.43666666666666665 * Signo(1.0 *
t_chapa - t_amb) * (t_amb - 1.0 * t_chapa)) / Abs(t_amb - 1.0 * t_chapa) ^ (2 /
3) + 9.0) + 0.0000002272 * epsilon_IR_chapa * t_chapa ^ 3 * (p - 1.0) + (4.0 *
t_chapa ^ 3 * (0.000000568 * p - 0.000000568)) / (1 / epsilon_IR_chapa + 1 /
epsilon_IR_vidrio - 1.0)

  J(0, 1) = -1.0 * (p - 1.0) * (0.38 * Abs(t_chapa - 1.0 * t_vidrio) ^
(1 / 3) + (0.50666666666666671 * Signo(1.0 * t_vidrio - t_chapa) * (0.5 * t_amb -
0.75 * t_chapa + 0.25 * t_vidrio)) / Abs(t_chapa - 1.0 * t_vidrio) ^ (2 / 3)) -
(4.0 * t_vidrio ^ 3 * (0.000000568 * p - 0.000000568)) / (1 / epsilon_IR_chapa
+ 1 / epsilon_IR_vidrio - 1.0)

  J(1, 0) = 0.38 * Abs(t_chapa - 1.0 * t_vidrio) ^ (1 / 3) -
(0.50666666666666671 * Signo(1.0 * t_vidrio - t_chapa) * (0.5 * t_amb + 0.25 *
t_chapa - 0.75 * t_vidrio)) / Abs(t_chapa - 1.0 * t_vidrio) ^ (2 / 3) - (4.0 *
t_chapa ^ 3 * (0.000000568 * p - 0.000000568)) / (1 / epsilon_IR_chapa + 1 /
epsilon_IR_vidrio - 1.0)

```

```

    J(1, 1) = (0.50666666666666671 * Signo(1.0 * t_vidrio - t_chapa) *
    (0.5 * t_amb + 0.25 * t_chapa - 0.75 * t_vidrio)) / Abs(t_chapa - 1.0 * t_vidrio)
    ^ (2 / 3) - 1.14 * Abs(t_chapa - 1.0 * t_vidrio) ^ (1 / 3) - 1.31 * Abs(t_int -
    1.0 * t_vidrio) ^ (1 / 3) - 0.0000002272 * epsilon_IR_vidrio * p * t_vidrio ^ 3 -
    0.000000168128 * epsilon_IR_vidrio * t_vidrio ^ 3 + (0.43666666666666665 *
    Signo(1.0 * t_vidrio - t_int) * (t_int - 1.0 * t_vidrio)) / Abs(t_int - 1.0 *
    t_vidrio) ^ (2 / 3) + (4.0 * t_vidrio ^ 3 * (0.0000000568 * p - 0.0000000568)) /
    (1 / epsilon_IR_chapa + 1 / epsilon_IR_vidrio - 1.0)
  
```

```

    ' Resolución del sistema linealizado
    I2 = {{-I2(0, 0)}, {-I2(1, 0)}}
    x1 = Linsolve_LU(J, I2)
    x0 = {{x0(0, 0) + x1(0, 0)}, {x0(1, 0) + x1(1, 0)}, {320}}
    errabs_chapa = x1(0, 0)
    errabs_vidrio = x1(1, 0)
    iter = iter + 1
  
```

Loop

```

    If iter <= 1000 Then
      x = x0
    Else
      x = {{Cdbl(iter)}, {Cdbl(iter)}, {0}}
    End If
  
```

```

    x(2, 0) = ((x(0, 0) + x(1, 0)) / 2 + t_amb) / 2
  
```

```

    Return x
  End Function
  
```

```

    Public Function Calcular_Ilu_Heat_AC(ByVal ciudad As String,
      ByVal material As String,
      ByVal p As Double,
      ByVal separacion As Double) As String()
  
```

```

    ' Las expresiones para el cálculo de los valores de las cargas vienen de
    una aproximación polinómica
    ' de cuarto grado de los valores de las tablas de Excel.
    ' ¡¡¡OJO!!! IMPORTANTISIMO hacer la correlación para valores de p en
    tanto por uno, si no da problemas de
    ' precisión por no usar suficientes decimales.
  
```

```

    Dim carga_y_ahorro(5) As String
    Dim sin_chapa_bilbao(2) As Double
    Dim sin_chapa_sevilla(2) As Double
    carga_y_ahorro = {"-----", "-----", "-----", "-----", "-----", "-----"},
    sin_chapa_bilbao = {845.6753, 1832.949, 4088.073} ' {ilum, heat, AC}
    sin_chapa_sevilla = {813.4565, 97.28376, 11780.18}
  
```

```

    Select Case material
      Case "Clean metallic galvanized sheet"
        If ((ciudad = "Bilbao") And (separacion = 5)) Then
  
```

```

      carga_y_ahorro(0) = CStr(Round(8650.3 * p ^ 4 - 17853 * p ^ 3
+ 13953 * p ^ 2 - 5168 * p + 1722, 3))
      carga_y_ahorro(1) = CStr(Round(2030.8 * p ^ 4 - 4493 * p ^ 3
+ 4452.3 * p ^ 2 - 3666.2 * p + 3442.6, 3))
      carga_y_ahorro(2) = CStr(Round(1502.5 * p ^ 4 - 3359.7 * p ^
3 + 4169.6 * p ^ 2 + 1262.6 * p + 841.13, 3))

      ElseIf ((ciudad = "Bilbao") And (separacion = 30)) Then
      carga_y_ahorro(0) = CStr(Round(8650.3 * p ^ 4 - 17853 * p ^ 3
+ 13953 * p ^ 2 - 5168 * p + 1722, 3))
      carga_y_ahorro(1) = CStr(Round(-2648.6 * p ^ 4 + 3885.9 * p ^
3 - 455.24 * p ^ 2 - 3076.4 * p + 3801.4, 3))
      carga_y_ahorro(2) = CStr(Round(1315.3 * p ^ 4 - 3394.9 * p ^
3 + 4851.3 * p ^ 2 + 860.86 * p + 715.31, 3))

      ElseIf ((ciudad = "Sevilla") And (separacion = 5)) Then
      carga_y_ahorro(0) = CStr(Round(22427 * p ^ 4 - 43302 * p ^ 3
+ 30448 * p ^ 2 - 9399.2 * p + 1971.9, 3))
      carga_y_ahorro(1) = CStr(Round(12288 * p ^ 4 - 24313 * p ^ 3
+ 18547 * p ^ 2 - 7382.2 * p + 1709.8, 3))
      carga_y_ahorro(2) = CStr(Round(37095 * p ^ 4 - 70671 * p ^ 3
+ 50225 * p ^ 2 - 7061.1 * p + 4209.2, 3))

      ElseIf ((ciudad = "Sevilla") And (separacion = 30)) Then
      carga_y_ahorro(0) = CStr(Round(14197 * p ^ 4 - 27664 * p ^ 3
+ 19769 * p ^ 2 - 6310.7 * p + 1655.1, 3))
      carga_y_ahorro(1) = CStr(Round(2377.1 * p ^ 4 - 5988.1 * p ^
3 + 6787.6 * p ^ 2 - 4613.2 * p + 1668.3, 3))
      carga_y_ahorro(2) = CStr(Round(37095 * p ^ 4 - 70671 * p ^ 3
+ 50225 * p ^ 2 - 7061.1 * p + 4209.2, 3))

      End If

      Case "Aluminium anodized"
      If ((ciudad = "Bilbao") And (separacion = 5)) Then
      carga_y_ahorro(0) = CStr(Round(8583.3 * p ^ 4 - 17703 * p ^ 3
+ 13819 * p ^ 2 - 5106.5 * p + 1708.5, 3))
      carga_y_ahorro(1) = CStr(Round(396.54 * p ^ 4 - 2181.5 * p ^
3 + 4178.2 * p ^ 2 - 4829.5 * p + 4063.9, 3))
      carga_y_ahorro(2) = CStr(Round(1158.2 * p ^ 4 - 2951.3 * p ^
3 + 4082.4 * p ^ 2 + 1362.7 * p + 752.54, 3))

      ElseIf ((ciudad = "Bilbao") And (separacion = 30)) Then
      carga_y_ahorro(0) = CStr(Round(8583.3 * p ^ 4 - 17703 * p ^ 3
+ 13819 * p ^ 2 - 5106.5 * p + 1708.5, 3))
      carga_y_ahorro(1) = CStr(Round(-1257.7 * p ^ 4 + 1224.8 * p ^
3 + 1825.3 * p ^ 2 - 4387.8 * p + 4232.1, 3))
      carga_y_ahorro(2) = CStr(Round(1586.9 * p ^ 4 - 3779.2 * p ^
3 + 4868.9 * p ^ 2 + 1002.8 * p + 718.69, 3))

      ElseIf ((ciudad = "Sevilla") And (separacion = 5)) Then
      carga_y_ahorro(0) = CStr(Round(22258 * p ^ 4 - 42971 * p ^ 3
+ 30209 * p ^ 2 - 9319.6 * p + 1960.1, 3))

```

```

        carga_y_ahorro(1) = CStr(Round(19712 * p ^ 4 - 38869 * p ^ 3
+ 29196 * p ^ 2 - 11043 * p + 2282.9, 3))
        carga_y_ahorro(2) = CStr(Round(13436 * p ^ 4 - 25712 * p ^ 3
+ 19433 * p ^ 2 + 2096.1 * p + 3152.8, 3))

        ElseIf ((ciudad = "Sevilla") And (separacion = 30)) Then
            carga_y_ahorro(0) = CStr(Round(27444 * p ^ 4 - 52823 * p ^ 3
+ 36937 * p ^ 2 - 11265 * p + 2159.7, 3))
            carga_y_ahorro(1) = CStr(Round(14988 * p ^ 4 - 29845 * p ^ 3
+ 23108 * p ^ 2 - 9493.9 * p + 2257.1, 3))
            carga_y_ahorro(2) = CStr(Round(13436 * p ^ 4 - 25712 * p ^ 3
+ 19433 * p ^ 2 + 2096.1 * p + 3152.8, 3))

        End If
        Case "Matte black painting"
            If ((ciudad = "Bilbao") And (separacion = 5)) Then
                carga_y_ahorro(0) = CStr(Round(8738.3 * p ^ 4 - 18070 * p ^ 3
+ 14168 * p ^ 2 - 5276.6 * p + 1747.7, 3))
                carga_y_ahorro(1) = CStr(Round(86.584 * p ^ 4 - 791.77 * p ^ 3
+ 1943.1 * p ^ 2 - 3200.4 * p + 3617.3, 3))
                carga_y_ahorro(2) = CStr(Round(497.45 * p ^ 4 - 1168.7 * p ^ 3
+ 2536.7 * p ^ 2 + 1485.8 * p + 973.45, 3))

                ElseIf ((ciudad = "Bilbao") And (separacion = 30)) Then
                    carga_y_ahorro(0) = CStr(Round(8738.3 * p ^ 4 - 18070 * p ^ 3
+ 14168 * p ^ 2 - 5276.6 * p + 1747.7, 3))
                    carga_y_ahorro(1) = CStr(Round(-2688.1 * p ^ 4 + 4696.9 * p ^ 3
- 1781.7 * p ^ 2 - 2466.7 * p + 3822.2, 3))
                    carga_y_ahorro(2) = CStr(Round(1463.8 * p ^ 4 - 3131 * p ^ 3
+ 4249.2 * p ^ 2 + 838.41 * p + 899.9, 3))

                    ElseIf ((ciudad = "Sevilla") And (separacion = 5)) Then
                        carga_y_ahorro(0) = CStr(Round(22761 * p ^ 4 - 43954 * p ^ 3
+ 30921 * p ^ 2 - 9557.2 * p + 1995.4, 3))
                        carga_y_ahorro(1) = CStr(Round(18089 * p ^ 4 - 35193 * p ^ 3
+ 25809 * p ^ 2 - 9491.3 * p + 1990.9, 3))
                        carga_y_ahorro(2) = CStr(Round(11102 * p ^ 4 - 20578 * p ^ 3
+ 15892 * p ^ 2 + 1981.7 * p + 3776.7, 3))

                        ElseIf ((ciudad = "Sevilla") And (separacion = 30)) Then
                            carga_y_ahorro(0) = CStr(Round(18645 * p ^ 4 - 36135 * p ^ 3
+ 25582 * p ^ 2 - 8012.9 * p + 1837, 3))
                            carga_y_ahorro(1) = CStr(Round(8555.9 * p ^ 4 - 16992 * p ^ 3
+ 13435 * p ^ 2 - 6168 * p + 1815.7, 3))
                            carga_y_ahorro(2) = CStr(Round(11102 * p ^ 4 - 20578 * p ^ 3
+ 15892 * p ^ 2 + 1981.7 * p + 3776.7, 3))

                            End If
                            Case "Glossy white painting"
                                If ((ciudad = "Bilbao") And (separacion = 5)) Then
                                    carga_y_ahorro(0) = CStr(Round(8499.7 * p ^ 4 - 17496 * p ^ 3
+ 13617 * p ^ 2 - 5007 * p + 1685.8, 3))
                                    carga_y_ahorro(1) = CStr(Round(-18675 * p ^ 4 + 33413 * p ^ 3
- 19019 * p ^ 2 + 1038.2 * p + 3716.9, 3))

```

```

      carga_y_ahorro(2) = CStr(Round(11968 * p ^ 4 - 24384 * p ^ 3
+ 19155 * p ^ 2 - 2818.8 * p + 1032, 3))

      ElseIf ((ciudad = "Bilbao") And (separacion = 30)) Then
        carga_y_ahorro(0) = CStr(Round(8499.7 * p ^ 4 - 17496 * p ^ 3
+ 13617 * p ^ 2 - 5007 * p + 1685.8, 3))
        carga_y_ahorro(1) = CStr(Round(-16840 * p ^ 4 + 30144 * p ^ 3
- 16812 * p ^ 2 + 251.43 * p + 3962.2, 3))
        carga_y_ahorro(2) = CStr(Round(8075.5 * p ^ 4 - 17609 * p ^ 3
+ 14996 * p ^ 2 - 1805.3 * p + 906.48, 3))

      ElseIf ((ciudad = "Sevilla") And (separacion = 5)) Then
        carga_y_ahorro(0) = CStr(Round(21994 * p ^ 4 - 42449 * p ^ 3
+ 29827 * p ^ 2 - 9190.4 * p + 1940.5, 3))
        carga_y_ahorro(1) = CStr(Round(1193.1 * p ^ 4 - 4314.3 * p ^
3 + 6488.4 * p ^ 2 - 5018.1 * p + 1791.4, 3))
        carga_y_ahorro(2) = CStr(Round(43383 * p ^ 4 - 84493 * p ^ 3
+ 60194 * p ^ 2 - 9028.9 * p + 3900.5, 3))

      ElseIf ((ciudad = "Sevilla") And (separacion = 30)) Then
        carga_y_ahorro(0) = CStr(Round(21994 * p ^ 4 - 42449 * p ^ 3
+ 29827 * p ^ 2 - 9190.4 * p + 1940.5, 3))
        carga_y_ahorro(1) = CStr(Round(2847.7 * p ^ 4 - 7365.4 * p ^
3 + 8565.9 * p ^ 2 - 5744 * p + 1967.4, 3))
        carga_y_ahorro(2) = CStr(Round(43383 * p ^ 4 - 84493 * p ^ 3
+ 60194 * p ^ 2 - 9028.9 * p + 3900.5, 3))

      End If

    End Select

    If Not (carga_y_ahorro(0) = "-----") And (ciudad = "Bilbao") Then
      carga_y_ahorro(3) = CStr(Round((sin_chapa_bilbao(0) -
carga_y_ahorro(0)) / sin_chapa_bilbao(0) * 100, 2))
      carga_y_ahorro(4) = CStr(Round(carga_y_ahorro(1) /
sin_chapa_bilbao(1) * 100, 2))
      carga_y_ahorro(5) = CStr(Round((sin_chapa_bilbao(2) -
carga_y_ahorro(2)) / sin_chapa_bilbao(2) * 100, 2))
      ElseIf Not (carga_y_ahorro(0) = "-----") And (ciudad = "Sevilla")
Then
        carga_y_ahorro(3) = CStr(Round((sin_chapa_sevilla(0) -
carga_y_ahorro(0)) / sin_chapa_sevilla(0) * 100, 2))
        carga_y_ahorro(4) = CStr(Round(carga_y_ahorro(1) /
sin_chapa_sevilla(1) * 100, 2))
        carga_y_ahorro(5) = CStr(Round((sin_chapa_sevilla(2) -
carga_y_ahorro(2)) / sin_chapa_sevilla(2) * 100, 2))
      End If

    If (FormPrincipal.checkSinChapa.Checked = True) And (ciudad = "Bilbao")
Then
      For i = 0 To 2
        carga_y_ahorro(i) = CStr(sin_chapa_bilbao(i))
        carga_y_ahorro(i + 3) = "0"
      Next
    
```



```

    ElseIf (FormPrincipal.checkSinChapa.Checked = True) And (ciudad =
"Sevilla") Then
      For i = 0 To 2
        carga_y_ahorro(i) = CStr(sin_chapa_sevilla(i))
        carga_y_ahorro(i + 3) = "0"
      Next
    End If

    Return carga_y_ahorro
  End Function

Public Sub Hacer_calculos()
  Dim temp(2, 0) As Double
  Dim x0(2, 0) As Double

  ' Cálculo de las temperaturas de la chapa, el vidrio y la cámara
  x0 = {{330}, {310}, {320}}
  temp = Calcular_tchapa_tvidrio(x0, epsilon_IR_chapa, epsilon_IR_vidrio,
h_camara,
                                p, t_amb, t_int, alpha_solar_chapa,
alpha_solar_vidrio,
                                Ir_sol)
  temp = {{temp(0, 0) - 273}, {temp(1, 0) - 273}, {temp(2, 0) - 273}}
  FormPrincipal.lblTchapa.Text = "T_chapa = " & CStr(Round(temp(0, 0), 3))
& " ° C"
  FormPrincipal.lblTvidrio.Text = "T_vidrio = " & CStr(Round(temp(1, 0),
3)) & " ° C"
  FormPrincipal.lblTgap.Text = "T_cámara = " & CStr(Round(temp(2, 0), 3)) &
" ° C"

  ' Cálculo del calor a aportar por el sistema de ACS en W/m²
  Dim epsilon_IR_interior As Double = 0.9
  Dim F_gnd As Double = 0.5
  Dim F_sky As Double = 0.5 * (0.5 ^ (1 / 2))
  Dim F_air As Double = 0.5 * (1 - 0.5 ^ (1 / 2))
  Dim t_ground As Double = t_amb
  Dim t_sky As Double = t_amb
  Dim t_air As Double = t_amb
  Dim t_chapa As Double = temp(0, 0) + 273
  Dim t_vidrio As Double = temp(1, 0) + 273

  Dim h_int As Double = 1.31 * (Abs(t_vidrio - t_int)) ^ (1 / 3)

  Dim qint_c As Double = h_int * (-t_int + t_vidrio)
  Dim qint_sw As Double = Ir_sol * p * tr_sw_vidrio
  Dim qint_lw_gnd As Double = epsilon_IR_interior * sigma * F_gnd *
(t_ground ^ 4 - t_int ^ 4)
  Dim qint_lw_sky As Double = epsilon_IR_interior * sigma * F_sky * (t_sky
^ 4 - t_int ^ 4)
  Dim qint_lw_air As Double = epsilon_IR_interior * sigma * F_air * (t_air
^ 4 - t_int ^ 4)

  Dim qint_lw_int_vidrio As Double = epsilon_IR_vidrio * sigma * (-t_int ^
4 + t_vidrio ^ 4)

```

```

Dim qint_lw_int_chapa As Double = epsilon_IR_chapa * sigma * (-t_int ^ 4
+ t_chapa ^ 4) * tr_lw_vidrio * (1 - p)

Dim qint_lw As Double = (qint_lw_air + qint_lw_sky + qint_lw_gnd) * p *
tr_lw_vidrio + qint_lw_int_vidrio + qint_lw_int_chapa

Dim HVAC As Double = -qint_c - qint_lw - qint_sw

FormPrincipal.lblHVAC.Visible = True

If HVAC > 0 Then
    FormPrincipal.lblHVAC.ForeColor = Color.Red
ElseIf HVAC = 0 Then
    FormPrincipal.lblHVAC.ForeColor = Color.Green
Else
    FormPrincipal.lblHVAC.ForeColor = Color.Blue
End If

FormPrincipal.lblHVAC.Text = CStr(Round(HVAC, 3))

' Cálculo de las cargas y el ahorro de iluminación, calefacción y aire
acondicionado
Dim demanda(5) As String
Dim cargas(2) As String
Dim ahorro(2) As String
Dim suma_cargas As String
Dim total_sin_chapa_bilbao As Double
Dim total_sin_chapa_sevilla As Double
demanda = Calcular_Ilu_Heat_AC(ciudad, material, p, separacion)
total_sin_chapa_bilbao = 6766.6973
total_sin_chapa_sevilla = 12690.92

For i = 0 To 2
    cargas(i) = demanda(i)
    ahorro(i) = demanda(i + 3)
Next

If Not (cargas(0) = "-----") Then
    FormConsumoAhorro.lviConsumoAhorro.Items(1).SubItems.Clear()
    FormConsumoAhorro.lviConsumoAhorro.Items(1).Text = "ILUMINACIÓN"
    FormConsumoAhorro.lviConsumoAhorro.Items(1).SubItems.Add(cargas(0) &
" kWh")
    FormConsumoAhorro.lviConsumoAhorro.Items(1).SubItems.Add(ahorro(0) &
" %")
    FormConsumoAhorro.lviConsumoAhorro.Items(3).SubItems.Clear()
    FormConsumoAhorro.lviConsumoAhorro.Items(3).Text = "CALEFACCIÓN"
    FormConsumoAhorro.lviConsumoAhorro.Items(3).SubItems.Add(cargas(1) &
" kWh")
    FormConsumoAhorro.lviConsumoAhorro.Items(3).SubItems.Add(ahorro(1) &
" % **")
    FormConsumoAhorro.lviConsumoAhorro.Items(5).SubItems.Clear()
    FormConsumoAhorro.lviConsumoAhorro.Items(5).Text = "AIRE
ACONDICIONADO"

```

```

FormConsumoAhorro.lviConsumoAhorro.Items(5).SubItems.Add(cargas(2) &
" kWh")
FormConsumoAhorro.lviConsumoAhorro.Items(5).SubItems.Add(ahorro(2) &
" %")

suma_cargas = CStr(CDb1(cargas(0)) + CDb1(cargas(1)) +
CDbl(cargas(2)))
FormConsumoAhorro.lviConsumoAhorro.Items(7).SubItems.Clear()
FormConsumoAhorro.lviConsumoAhorro.Items(7).Text = "TOTAL"
FormConsumoAhorro.lviConsumoAhorro.Items(7).SubItems.Add(suma_cargas
& " kWh")
If FormPrincipal.cboCity.Text = "Bilbao" Then

FormConsumoAhorro.lviConsumoAhorro.Items(7).SubItems.Add(CStr(Round((total_sin_ch
apa_bilbao - CDb1(suma_cargas)) / total_sin_chapa_bilbao * 100, 2)) & " %")
ElseIf FormPrincipal.cboCity.Text = "Sevilla" Then

FormConsumoAhorro.lviConsumoAhorro.Items(7).SubItems.Add(CStr(Round((total_sin_ch
apa_sevilla - CDb1(suma_cargas)) / total_sin_chapa_sevilla * 100, 2)) & " %")
End If

Else
FormConsumoAhorro.lviConsumoAhorro.Items(1).SubItems.Clear()
FormConsumoAhorro.lviConsumoAhorro.Items(1).Text = "ILUMINACIÓN"
FormConsumoAhorro.lviConsumoAhorro.Items(1).SubItems.Add("No hay
datos")
FormConsumoAhorro.lviConsumoAhorro.Items(1).SubItems.Add("No hay
datos")
FormConsumoAhorro.lviConsumoAhorro.Items(3).SubItems.Clear()
FormConsumoAhorro.lviConsumoAhorro.Items(3).Text = "CALEFACCIÓN"
FormConsumoAhorro.lviConsumoAhorro.Items(3).SubItems.Add("No hay
datos")
FormConsumoAhorro.lviConsumoAhorro.Items(3).SubItems.Add("No hay
datos")
FormConsumoAhorro.lviConsumoAhorro.Items(5).SubItems.Clear()
FormConsumoAhorro.lviConsumoAhorro.Items(5).Text = "AIRE
ACONDICIONADO"
FormConsumoAhorro.lviConsumoAhorro.Items(5).SubItems.Add("No hay
datos")
FormConsumoAhorro.lviConsumoAhorro.Items(5).SubItems.Add("No hay
datos")
FormConsumoAhorro.lviConsumoAhorro.Items(7).SubItems.Clear()
FormConsumoAhorro.lviConsumoAhorro.Items(7).Text = "TOTAL"
FormConsumoAhorro.lviConsumoAhorro.Items(7).SubItems.Add("")
FormConsumoAhorro.lviConsumoAhorro.Items(7).SubItems.Add("")

End If

End Sub

Public Function Determinar_Ir_Tamb_Ciudad(ByVal ciudad As String,
ByVal mes As String) As Double()
' Función que determina el valor de la irradiancia y la temperatura
ambiente medios en función

```

```
' de la ciudad y la época del año
,
' Cada fila de las matrices son datos de una ciudad concreta y cada
columna es un mes distinto.
' Los datos está en una hoja de Excel de nombre Datos Climatológicos.xlsx
Dim irrad_tamb(1) As Double
Dim matIrradiancia(25, 11) As Double
Dim matTamb(25, 11) As Double
Dim indice_irrad(1) As Integer
Dim indice_tamb(1) As Integer
Dim ciudades(25) As String
Dim meses(11) As String

matIrradiancia = {{247.5, 220.0, 189.58, 149.58, 116.25, 105.0, 113.33,
150.42, 193.75, 230.83, 246.25, 261.67},
{264.58, 243.33, 191.25, 137.08, 95.42, 77.08, 85.0,
115.83, 154.58, 201.67, 237.92, 259.17},
{109.58, 128.75, 147.08, 178.33, 201.67, 190.83,
215.42, 199.17, 169.58, 140.0, 116.25, 107.5},
{116.25, 153.75, 196.25, 239.58, 257.08, 238.33,
213.75, 194.58, 177.5, 152.92, 117.5, 102.5},
{132.08, 124.58, 130.0, 155.42, 179.17, 189.17, 207.08,
190.0, 180.83, 176.25, 157.92, 130.42},
{109.58, 120.83, 133.75, 167.92, 187.92, 180.83,
217.08, 196.67, 161.25, 140.42, 116.25, 111.25},
{125.83, 115.83, 126.25, 151.25, 176.67, 188.75,
206.67, 192.92, 180.83, 175.83, 157.92, 135.42},
{340.42, 304.17, 246.25, 179.17, 128.75, 110.0, 118.75,
152.92, 205.0, 269.17, 320.0, 340.83},
{275.83, 256.67, 231.67, 203.33, 185.0, 168.75, 182.5,
212.08, 247.92, 253.33, 270.0, 279.17},
{225.42, 223.33, 203.33, 189.58, 182.08, 162.92,
142.08, 126.67, 143.33, 154.17, 179.17, 210.42},
{267.5, 285.83, 277.5, 242.92, 223.33, 212.92, 217.92,
231.25, 265.42, 255.42, 232.92, 252.5},
{202.5, 201.25, 204.58, 193.75, 196.67, 201.25, 208.33,
211.25, 209.58, 195.83, 191.25, 191.25},
{217.08, 225.42, 198.33, 179.17, 152.5, 152.08, 152.5,
180.0, 174.17, 197.5, 207.08, 208.75},
{168.75, 206.25, 224.58, 271.67, 305.0, 309.58, 276.67,
270.0, 257.5, 233.33, 192.92, 157.08},
{211.25, 234.17, 249.58, 259.58, 223.75, 205.42,
200.83, 200.42, 197.08, 187.08, 197.5, 202.5},
{158.33, 195.0, 241.67, 262.5, 267.5, 252.92, 217.5,
200.42, 210.42, 201.25, 174.17, 146.67},
{221.67, 260.42, 293.75, 307.5, 305.42, 235.0, 208.33,
213.33, 235.42, 238.33, 224.17, 208.33},
{43.33, 72.08, 115.83, 164.58, 210.0, 224.58, 223.33,
199.58, 141.25, 85.0, 50.0, 34.58},
{32.08, 57.92, 97.5, 149.58, 190.42, 201.67, 200.0,
176.25, 119.17, 72.08, 40.0, 25.0},
{16.67, 44.58, 101.67, 170.0, 233.33, 244.17, 232.92,
182.5, 117.92, 55.42, 22.92, 10.42},
```

{80.42, 115.83, 155.42, 191.25, 218.75, 236.25, 231.67,
207.92, 175.83, 131.67, 87.08, 69.58},
{123.75, 155.83, 215.83, 279.17, 313.75, 328.33,
315.83, 287.5, 241.25, 186.67, 142.08, 115.0},
{65.0, 92.92, 142.92, 179.17, 215.42, 231.25, 228.75,
202.92, 170.0, 113.33, 70.83, 57.5},
{87.5, 133.33, 200.0, 245.83, 291.67, 333.33, 325.0,
287.5, 220.83, 150.0, 100.0, 75.0},
{113.33, 152.5, 209.58, 255.83, 291.25, 328.33, 337.5,
300.0, 240.83, 167.5, 121.67, 97.08},
{108.33, 145.83, 195.83, 250.0, 308.33, 325.0, 333.33,
270.83, 225.0, 154.17, 108.33, 91.67}}

matTamb = {{23.05, 23.1, 21.75, 19.3, 16.55, 13.85, 13.05, 14.3, 16.6,
18.6, 20.05, 21.9},
{20.95, 21.3, 19.45, 16.4, 13.65, 11.35, 10.65, 11.75, 13.45,
15.35, 17.6, 19.25},
{4.1, 5.05, 8.9, 14.6, 19.6, 23.75, 28.2, 28.1, 23.85, 18.5,
12.65, 6.5},
{-3.9, -1.45, 5.35, 13.55, 19.8, 24.25, 26.2, 24.95, 20, 13.15,
4.85, -1.8},
{16.55, 16.95, 19.3, 22.9, 26.25, 28.2, 29.1, 28.85, 27.95,
25.75, 21.95, 18.05},
{14.05, 14.85, 18.25, 22.3, 26.05, 27.9, 29, 28.9, 27.6, 24.55,
20.05, 15.8},
{14.95, 15.4, 18.45, 22.35, 25.85, 28, 28.9, 28.6, 27.45,
24.85, 20.6, 16.7},
{20.9, 21.05, 19.8, 17.45, 14.85, 12.95, 12.25, 12.65, 13.95,
15.95, 18.35, 19.9},
{20.15, 19.6, 18.55, 15.7, 13.05, 10.05, 10.4, 12.8, 16.05,
17.5, 18.45, 19.55},
{27.8, 28.45, 28.2, 27.8, 27.3, 26.45, 25.4, 25.25, 25.9,
26.15, 27.05, 27.5},
{18, 18.6, 19.35, 19.05, 17.9, 16.25, 15.35, 15.8, 17.1, 18.6,
18.1, 18},
{11, 11.6, 12.15, 12.3, 12.45, 12, 11.3, 11.35, 11.45, 11.8,
11.95, 11.2},
{26.35, 26.85, 26.35, 24.85, 23.4, 21.95, 21.85, 22.25, 22.1,
23.1, 24.4, 25.5},
{18.85, 19.55, 22.45, 26.25, 30.15, 32.45, 34.75, 34.85, 32.5,
28.9, 24.4, 20.8},
{26.5, 28, 29.3, 30.5, 29.8, 29.25, 28.85, 28.7, 28.45, 28.15,
27.35, 26.05},
{14.3, 17, 22.5, 28.75, 33, 33.85, 31.1, 29.9, 29.35, 26.15,
20.4, 15.55},
{23.7, 24.45, 26.75, 28.4, 29.85, 29.15, 27.55, 27.2, 27.3,
28.25, 27.15, 25.2},
{4.7, 5.5, 8.45, 10.75, 14.75, 17.55, 19.95, 20, 16.65, 12.5,
7.85, 5.7},
{5.6, 5.65, 8.1, 10.25, 13.5, 16.4, 18.55, 18.45, 15.7, 12.15,
8.55, 5.9},
{-3, -3, 0, 5, 11, 16, 17.5, 16.5, 12, 7.5, 3, -1},
{0.35, 1.8, 5.8, 11.65, 16.9, 21.95, 24.65, 24, 20, 13.85, 8.7,
3.05},

```

    {14.45, 14.95, 15.9, 17.25, 18.75, 20.65, 23, 23.45, 22.85,
    20.3, 16.9, 14.2},
    {9.3, 9.7, 11.5, 12.6, 15.7, 18.4, 20.4, 20.9, 19.2, 16.4,
    12.4, 9.9},
    {6.25, 7.85, 11.25, 12.95, 16.75, 22.15, 25.55, 25.05, 20.9,
    15.05, 9.85, 6.8},
    {10.9, 12.5, 15.6, 17.3, 20.7, 25.1, 28.2, 27.9, 25, 20.2,
    15.1, 11.9},
    {11.75, 12.45, 14.45, 16.15, 19, 22.85, 25.6, 26.05, 23.5,
    19.75, 15.3, 12.55}}
  
```

```

    ciudades = {"Sidney", "Melbourne", "Shangai", "Beijing", "Hong Kong",
    "Guangzhou", "Macau", "Cape Town",
    "Johannesburgo", "Douala", "Nairobi", "Bogotá", "Rio de
    Janeiro", "Dubai", "Bangkok", "Nueva Delhi",
    "Bombay", "París", "Londres", "Estocolmo", "Nueva York", "Los
    Ángeles", "Bilbao", "Madrid", "Sevilla",
    "Valencia"}
  
```

```

    meses = {"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio",
    "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"}
  
```

```

    For i = 0 To (ciudades.Length - 1)
      If ciudades(i) = ciudad Then
        indice_irrad(0) = i
        indice_tamb(0) = i
        Exit For
      End If
    Next

    For i = 0 To (meses.Length - 1)
      If meses(i) = mes Then
        indice_irrad(1) = i
        indice_tamb(1) = i
        Exit For
      End If
    Next

    irrad_tamb(0) = matIrradiancia(indice_irrad(0), indice_irrad(1))
    irrad_tamb(1) = matTamb(indice_tamb(0), indice_tamb(1)) + 273

    Return irrad_tamb
  End Function
End Module
  
```

modGraficos

```
Imports System.Math
```

```
Module modGraficos
```

```
Public Sub Posicionar_barras(ByVal Ir_sol As Double,  
ByVal p As Double,  
ByVal t_amb As Double)
```

```
FormPrincipal.trackIrradiancia.Value = CInt(Round(Ir_sol))  
If (FormPrincipal.checkSinChapa.Checked = False) And  
(FormPrincipal.checkChapaMaciza.Checked = False) Then  
FormPrincipal.trackPerfRate.Value = CInt(Round(100 * p))  
End If  
FormPrincipal.trackTamb.Value = CInt(Round(t_amb - 273))  
  
FormPrincipal.txtBarraIrrad.Text =  
CStr(FormPrincipal.trackIrradiancia.Value) & " W/m²"  
FormPrincipal.txtBarraPerfRate.Text =  
CStr(FormPrincipal.trackPerfRate.Value) & "%"  
FormPrincipal.txtBarraTamb.Text = CStr(FormPrincipal.trackTamb.Value) & "  
° C"
```

```
End Sub
```

```
Private Function Calcular_HVAC_para_graficar(ByVal t_chapa As Double,  
ByVal t_vidrio As Double) As
```

```
Double
```

```
' Cálculo del calor a aportar por el sistema de ACS en W/m²  
Dim epsilon_IR_interior As Double = 0.9  
Dim F_gnd As Double = 0.5  
Dim F_sky As Double = 0.5 * (0.5 ^ (1 / 2))  
Dim F_air As Double = 0.5 * (1 - 0.5 ^ (1 / 2))  
Dim t_ground As Double = t_amb  
Dim t_sky As Double = t_amb  
Dim t_air As Double = t_amb  
  
Dim h_int As Double = 1.31 * (Abs(t_vidrio - t_int)) ^ (1 / 3)  
  
Dim qint_c As Double = h_int * (-t_int + t_vidrio)  
Dim qint_sw As Double = Ir_sol * p * tr_sw_vidrio  
Dim qint_lw_gnd As Double = epsilon_IR_interior * sigma * F_gnd *  
(t_ground ^ 4 - t_int ^ 4)  
Dim qint_lw_sky As Double = epsilon_IR_interior * sigma * F_sky * (t_sky  
^ 4 - t_int ^ 4)  
Dim qint_lw_air As Double = epsilon_IR_interior * sigma * F_air * (t_air  
^ 4 - t_int ^ 4)  
  
Dim qint_lw_int_vidrio As Double = epsilon_IR_vidrio * sigma * (-t_int ^ 4  
+ t_vidrio ^ 4)  
Dim qint_lw_int_chapa As Double = epsilon_IR_chapa * sigma * (-t_int ^ 4  
+ t_chapa ^ 4) * tr_lw_vidrio * (1 - p)
```

```

    Dim qint_lw As Double = (qint_lw_air + qint_lw_sky + qint_lw_gnd) * p *
    tr_lw_vidrio + qint_lw_int_vidrio + qint_lw_int_chapa
  
```

```

    Dim HVAC As Double = -qint_c - qint_lw - qint_sw
  
```

```

    Return HVAC
  End Function
  
```

```

  Public Function Incluye_nombre(nombre As String, array As String()) As
  Boolean
  
```

```

    Dim boEstaIncluido As Boolean = False
  
```

```

    For i = 0 To array.Length - 1
      If nombre = array(i) Then
        boEstaIncluido = True
      End If
    Next
  
```

```

    Return boEstaIncluido
  End Function
  
```

```

  Public Sub Generar_grafico(ByVal vble_x As String,
                             ByVal vbles_y As String(),
                             ByVal xmin As Integer,
                             ByVal xmax As Integer,
                             ByVal tipo_grafico As String,
                             ByVal titulo_grafico As String)
  
```

```

    ' Cálculo de los vectores que van a graficarse
  
```

```

    Dim strVariables(3) As String
    Dim dblVariables(2) As Double
    Dim temp(3, (xmax - xmin) + 1) As Double
    Dim comp_mat(15, (xmax - xmin) + 1) As Double
    Dim t(2, 0) As Double
    Dim x0(2, 0) As Double
    Dim hvac As Double
    Dim interval As Integer
    x0 = {{330}, {310}, {320}}
  
```

```

    ' Almaceno los valores que se muestran en pantalla en el momento de
    graficar
  
```

```

    strVariables(0) = FormPrincipal.lblTchapa.Text
    strVariables(1) = FormPrincipal.lblTvidrio.Text
    strVariables(2) = FormPrincipal.lblTgap.Text
    strVariables(3) = FormPrincipal.lblHVAC.Text
    dblVariables(0) = Ir_sol
    If FormPrincipal.checkSinChapa.Checked = True Then
      dblVariables(1) = 0.7
    ElseIf FormPrincipal.checkChapaMaciza.Checked = True Then
      dblVariables(1) = 0.1
    Else
      dblVariables(1) = p
    End If
    dblVariables(2) = t_amb
  
```



```

' Hago invisibles todas las etiquetas en las que se muestran valores para
que no se alteren al graficar
FormPrincipal.lblTchapa.Visible = False
FormPrincipal.lblTvidrio.Visible = False
FormPrincipal.lblTgap.Visible = False
FormPrincipal.lblHVAC.Visible = False

' Calculo las matrices que contendrán los valores que van a ser
graficados

If tipo_grafico = "column" Then
  interval = FormPrincipal.numIntervalo.Value
Else
  interval = 1
End If

Select Case vble_x
  Case "t_amb"
    For i = (xmin + 273) To (xmax + 273) Step interval
      t_amb = i
      t = Calcular_tchapa_tvidrio(x0, epsilon_IR_chapa,
epsilon_IR_vidrio, h_camara,
                                p, t_amb, t_int, alpha_solar_chapa,
                                alpha_solar_vidrio,
                                Ir_sol)
      hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
      temp(0, i - (xmin + 273)) = t(0, 0) - 273 ' t_chapa
      temp(1, i - (xmin + 273)) = t(1, 0) - 273 ' t_vidrio
      temp(2, i - (xmin + 273)) = t(2, 0) - 273 ' t_gap
      temp(3, i - (xmin + 273)) = hvac ' HVAC
    Next

  Case "Ir_sol"
    For i = xmin To xmax Step interval
      Ir_sol = i
      t = Calcular_tchapa_tvidrio(x0, epsilon_IR_chapa,
epsilon_IR_vidrio, h_camara,
                                p, t_amb, t_int, alpha_solar_chapa,
                                alpha_solar_vidrio,
                                Ir_sol)
      hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
      temp(0, i - xmin) = t(0, 0) - 273 ' t_chapa
      temp(1, i - xmin) = t(1, 0) - 273 ' t_vidrio
      temp(2, i - xmin) = t(2, 0) - 273 ' t_gap
      temp(3, i - xmin) = hvac ' HVAC
    Next

  Case "perf_ratio"
    For i = xmin To xmax Step interval
      p = i / 100
      t = Calcular_tchapa_tvidrio(x0, epsilon_IR_chapa,
epsilon_IR_vidrio, h_camara,

```

```

p, t_amb, t_int, alpha_solar_chapa,
alpha_solar_vidrio,
                                Ir_sol)
                                hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
                                temp(0, i - xmin) = t(0, 0) - 273 ' t_chapa
                                temp(1, i - xmin) = t(1, 0) - 273 ' t_vidrio
                                temp(2, i - xmin) = t(2, 0) - 273 ' t_gap
                                temp(3, i - xmin) = hvac ' HVAC
                                Next
                                End Select
                                ' Creo el gráfico
                                FormGraficos.chartGrafico.Series.Clear()
                                FormGraficos.chartGrafico.Legends.Clear()
                                FormGraficos.chartGrafico.Titles(0).Name = titulo_grafico
                                FormGraficos.chartGrafico.Titles(0).Font = New Font("Microsoft Sans
                                Serif", 18, FontStyle.Bold, GraphicsUnit.Pixel)
                                FormGraficos.chartGrafico.Titles(0).Text = titulo_grafico
                                If interval < 10 Then
                                    FormGraficos.chartGrafico.ChartAreas(0).AxisX.Interval = Round((xmax
                                - xmin) / 10, 0)
                                Else
                                    FormGraficos.chartGrafico.ChartAreas(0).AxisX.Interval = interval
                                End If
                                For i = 0 To vbles_y.Length - 1
                                    FormGraficos.chartGrafico.Series.Add(vbles_y(i))
                                Next
                                For i = 0 To vbles_y.Length - 1
                                    If tipo_grafico = "column" Then
                                        FormGraficos.chartGrafico.Series(i).ChartType =
                                DataVisualization.Charting.SeriesChartType.Column
                                    ElseIf tipo_grafico = "spline" Then
                                        FormGraficos.chartGrafico.Series(i).ChartType =
                                DataVisualization.Charting.SeriesChartType.Spline
                                    End If
                                    FormGraficos.chartGrafico.Series(i).Name = vbles_y(i)
                                    ' FormGraficos.chartGrafico.Series(i).MarkerStyle =
                                DataVisualization.Charting.MarkerStyle.Circle
                                    ' FormGraficos.chartGrafico.Series(i).MarkerSize = 5
                                    FormGraficos.chartGrafico.Series(i).BorderWidth = 3
                                    If (tipo_grafico = "column") And (interval > 15) Then
                                        FormGraficos.chartGrafico.Series(i).CustomProperties =
                                "PointWidth=" & CStr(interval - 6)
                                    ElseIf (tipo_grafico = "column") And (interval <= 15) Then
                                        FormGraficos.chartGrafico.Series(i).CustomProperties =
                                "PointWidth=1"
                                    End If

```

```

        If FormPrincipal.checkLeyendaGraficos.Checked = True Then
            FormGraficos.chartGrafico.Legends.Add(vbles_y(i))
            FormGraficos.chartGrafico.Legends(vbles_y(i)).Font      =      New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)
            FormGraficos.chartGrafico.Legends(vbles_y(i)).BackColor  =
SystemColors.Control
        End If

    Next

    Dim boYaIncluida(3) As Boolean
    boYaIncluida = {False, False, False, False}

    For i = 0 To vbles_y.Length - 1
        FormGraficos.chartGrafico.Series(i).Points.Clear()
        FormGraficos.chartGrafico.ChartAreas(0).AxisX.Title = vbles_x
        FormGraficos.chartGrafico.ChartAreas(0).AxisX.TitleFont      =      New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)
        FormGraficos.chartGrafico.ChartAreas(0).AxisY.Title = vbles_y(i)
        FormGraficos.chartGrafico.ChartAreas(0).AxisY.TitleFont      =      New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)

        If Incluye_nombre("t_chapa", vbles_y) And (boYaIncluida(0) = False)
Then
            For j = 0 To (xmax - xmin)
                FormGraficos.chartGrafico.Series(i).Points.AddXY(xmin + j,
temp(0, j))
            Next
            boYaIncluida(0) = True
            Continue For
        End If

        If Incluye_nombre("t_vidrio", vbles_y) And (boYaIncluida(1) = False)
Then
            For j = 0 To (xmax - xmin)
                FormGraficos.chartGrafico.Series(i).Points.AddXY(xmin + j,
temp(1, j))
            Next
            boYaIncluida(1) = True
            Continue For
        End If

        If Incluye_nombre("t_gap", vbles_y) And (boYaIncluida(2) = False)
Then
            For j = 0 To (xmax - xmin)
                FormGraficos.chartGrafico.Series(i).Points.AddXY(xmin + j,
temp(2, j))
            Next
            boYaIncluida(2) = True
            Continue For
        End If

        If Incluye_nombre("HVAC", vbles_y) And (boYaIncluida(3) = False) Then
            For j = 0 To (xmax - xmin)

```

```

        FormGraficos.chartGrafico.Series(i).Points.AddXY(xmin + j,
temp(3, j))
        Next
        boYaIncluida(3) = True
    End If

    FormGraficos.chartGrafico.ChartAreas(0).AxisX.MajorGrid.LineColor =
Color.LightGray
    FormGraficos.chartGrafico.ChartAreas(0).AxisX.MinorGrid.LineColor =
Color.LightGray
    FormGraficos.chartGrafico.ChartAreas(0).AxisY.MajorGrid.LineColor =
Color.LightGray
    FormGraficos.chartGrafico.ChartAreas(0).AxisY.MinorGrid.LineColor =
Color.LightGray

    Next

    ' Para finalizar, vuelvo a mostrar en pantalla los valores que había
antes de graficar
    FormPrincipal.lblTchapa.Visible = True
    FormPrincipal.lblTvidrio.Visible = True
    FormPrincipal.lblTgap.Visible = True
    FormPrincipal.lblHVAC.Visible = True
    Posicionar_barras(dblVariables(0), dblVariables(1), dblVariables(2))

End Sub

Private Function Valores_grafico_materiales(ByVal vble_x As String,
ByVal xmin As Integer,
ByVal xmax As Integer) As
Double(,)
    Dim comp_mat(15, (xmax - xmin) + 1) As Double
    Dim t(2, 0) As Double
    Dim hvac As Double
    Dim x0(2, 0) As Double
    x0 = {{330}, {310}, {320}}

    Select Case vble_x
        Case "t_amb"
            For i = (xmin + 273) To (xmax + 273)
                t_amb = i

                ' Aluminum anodized alpha_solar_chapa = 0.5
                epsilon_IR_chapa = 0.84
                t = Calcular_tchapa_tvidrio(x0, 0.84, epsilon_IR_vidrio,
h_camara,
                p, t_amb, t_int, 0.5,
                alpha_solar_vidrio, Ir_sol)
                hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
                comp_mat(0, i - (xmin + 273)) = t(0, 0) - 273 ' t_chapa
            for Aluminium anodized
                comp_mat(4, i - (xmin + 273)) = t(1, 0) - 273 ' t_vidrio
            for Aluminium anodized

```

```

comp_mat(8, i - (xmin + 273)) = t(2, 0) - 273      ' t_gap for
Aluminium anodized
comp_mat(12, i - (xmin + 273)) = hvac              ' HVAC for
Aluminium anodized

' Clean metallic galvanized sheet      alpha_solar_chapa =
0.65  epsilon_IR_chapa = 0.13
t = Calcular_tchapa_tvidrio(x0, 0.13, epsilon_IR_vidrio,
h_camara,
                                p,    t_amb,    t_int,    0.65,
alpha_solar_vidrio, Ir_sol)
hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
comp_mat(1, i - (xmin + 273)) = t(0, 0) - 273      ' t_chapa
for Clean metallic galvanized sheet
comp_mat(5, i - (xmin + 273)) = t(1, 0) - 273      ' t_vidrio
for Clean metallic galvanized sheet
comp_mat(9, i - (xmin + 273)) = t(2, 0) - 273      ' t_gap for
Clean metallic galvanized sheet
comp_mat(13, i - (xmin + 273)) = hvac              ' HVAC for
Clean metallic galvanized sheet

' Matte black painting                  alpha_solar_chapa = 0.95
epsilon_IR_chapa = 0.89
t = Calcular_tchapa_tvidrio(x0, 0.89, epsilon_IR_vidrio,
h_camara,
                                p,    t_amb,    t_int,    0.95,
alpha_solar_vidrio, Ir_sol)
hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
comp_mat(2, i - (xmin + 273)) = t(0, 0) - 273      ' t_chapa
for Matte black painting
comp_mat(6, i - (xmin + 273)) = t(1, 0) - 273      ' t_vidrio
for Matte black painting
comp_mat(10, i - (xmin + 273)) = t(2, 0) - 273     ' t_gap for
Matte black painting
comp_mat(14, i - (xmin + 273)) = hvac              ' HVAC for
Matte black painting

' Glossy white painting                  alpha_solar_chapa = 0.25
epsilon_IR_chapa = 0.89
t = Calcular_tchapa_tvidrio(x0, 0.89, epsilon_IR_vidrio,
h_camara,
                                p,    t_amb,    t_int,    0.25,
alpha_solar_vidrio, Ir_sol)
hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
comp_mat(3, i - (xmin + 273)) = t(0, 0) - 273      ' t_chapa
for Matte black painting
comp_mat(7, i - (xmin + 273)) = t(1, 0) - 273      ' t_vidrio
for Matte black painting
comp_mat(11, i - (xmin + 273)) = t(2, 0) - 273     ' t_gap for
Matte black painting
comp_mat(15, i - (xmin + 273)) = hvac              ' HVAC for
Matte black painting

```

Next

```

Case "Ir_sol"
  For i = xmin To xmax
    Ir_sol = i

    ' Aluminum anodized          alpha_solar_chapa = 0.5
    epsilon_IR_chapa = 0.84
    t = Calcular_tchapa_tvidrio(x0, 0.84, epsilon_IR_vidrio,
    h_camara,
                                p, t_amb, t_int, 0.5,
    alpha_solar_vidrio, Ir_sol)
    hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
    comp_mat(0, i - xmin) = t(0, 0) - 273 ' t_chapa for
    Aluminium anodized
    comp_mat(4, i - xmin) = t(1, 0) - 273 ' t_vidrio for
    Aluminium anodized
    comp_mat(8, i - xmin) = t(2, 0) - 273 ' t_gap for Aluminium
    anodized
    comp_mat(12, i - xmin) = hvac ' HVAC for Aluminium
    anodized

    ' Clean metallic galvanized sheet alpha_solar_chapa =
    0.65 epsilon_IR_chapa = 0.13
    t = Calcular_tchapa_tvidrio(x0, 0.13, epsilon_IR_vidrio,
    h_camara,
                                p, t_amb, t_int, 0.65,
    alpha_solar_vidrio, Ir_sol)
    hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
    comp_mat(1, i - xmin) = t(0, 0) - 273 ' t_chapa for Clean
    metallic galvanized sheet
    comp_mat(5, i - xmin) = t(1, 0) - 273 ' t_vidrio for Clean
    metallic galvanized sheet
    comp_mat(9, i - xmin) = t(2, 0) - 273 ' t_gap for Clean
    metallic galvanized sheet
    comp_mat(13, i - xmin) = hvac ' HVAC for Clean
    metallic galvanized sheet

    ' Matte black painting          alpha_solar_chapa = 0.95
    epsilon_IR_chapa = 0.89
    t = Calcular_tchapa_tvidrio(x0, 0.89, epsilon_IR_vidrio,
    h_camara,
                                p, t_amb, t_int, 0.95,
    alpha_solar_vidrio, Ir_sol)
    hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
    comp_mat(2, i - xmin) = t(0, 0) - 273 ' t_chapa for Matte
    black painting
    comp_mat(6, i - xmin) = t(1, 0) - 273 ' t_vidrio for Matte
    black painting
    comp_mat(10, i - xmin) = t(2, 0) - 273 ' t_gap for Matte
    black painting
    comp_mat(14, i - xmin) = hvac ' HVAC for Matte
    black painting
  
```

```

    ' Glossy white painting          alpha_solar_chapa = 0.25
epsilon_IR_chapa = 0.89
    t = Calcular_tchapa_tvidrio(x0, 0.89, epsilon_IR_vidrio,
h_camara,
                                p,    t_amb,    t_int,    0.25,
alpha_solar_vidrio, Ir_sol)
    hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
    comp_mat(3, i - xmin) = t(0, 0) - 273 ' t_chapa for Matte
black painting
    comp_mat(7, i - xmin) = t(1, 0) - 273 ' t_vidrio for Matte
black painting
    comp_mat(11, i - xmin) = t(2, 0) - 273 ' t_gap for Matte
black painting
    comp_mat(15, i - xmin) = hvac          ' HVAC for Matte
black painting
  
```

Next

Case "perf_ratio"

For i = xmin To xmax

p = i / 100

```

    ' Aluminum anodized          alpha_solar_chapa = 0.5
epsilon_IR_chapa = 0.84
    t = Calcular_tchapa_tvidrio(x0, 0.84, epsilon_IR_vidrio,
h_camara,
                                p,    t_amb,    t_int,    0.5,
alpha_solar_vidrio, Ir_sol)
    hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
    comp_mat(0, i - xmin) = t(0, 0) - 273 ' t_chapa for
Aluminium anodized
    comp_mat(4, i - xmin) = t(1, 0) - 273 ' t_vidrio for
Aluminium anodized
    comp_mat(8, i - xmin) = t(2, 0) - 273 ' t_gap for Aluminium
anodized
    comp_mat(12, i - xmin) = hvac          ' HVAC for Aluminium
anodized

    ' Clean metallic galvanized sheet alpha_solar_chapa =
0.65   epsilon_IR_chapa = 0.13
    t = Calcular_tchapa_tvidrio(x0, 0.13, epsilon_IR_vidrio,
h_camara,
                                p,    t_amb,    t_int,    0.65,
alpha_solar_vidrio, Ir_sol)
    hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
    comp_mat(1, i - xmin) = t(0, 0) - 273 ' t_chapa for Clean
metallic galvanized sheet
    comp_mat(5, i - xmin) = t(1, 0) - 273 ' t_vidrio for Clean
metallic galvanized sheet
    comp_mat(9, i - xmin) = t(2, 0) - 273 ' t_gap for Clean
metallic galvanized sheet
    comp_mat(13, i - xmin) = hvac          ' HVAC for Clean
metallic galvanized sheet
  
```

```

' Matte black painting          alpha_solar_chapa = 0.95
epsilon_IR_chapa = 0.89
t = Calcular_tchapa_tvidrio(x0, 0.89, epsilon_IR_vidrio,
h_camara,
                                p,    t_amb,    t_int,    0.95,
alpha_solar_vidrio, Ir_sol)
hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
comp_mat(2, i - xmin) = t(0, 0) - 273 ' t_chapa for Matte
black painting
comp_mat(6, i - xmin) = t(1, 0) - 273 ' t_vidrio for Matte
black painting
comp_mat(10, i - xmin) = t(2, 0) - 273 ' t_gap for Matte
black painting
comp_mat(14, i - xmin) = hvac          ' HVAC for Matte
black painting

' Glossy white painting        alpha_solar_chapa = 0.25
epsilon_IR_chapa = 0.89
t = Calcular_tchapa_tvidrio(x0, 0.89, epsilon_IR_vidrio,
h_camara,
                                p,    t_amb,    t_int,    0.25,
alpha_solar_vidrio, Ir_sol)
hvac = Calcular_HVAC_para_graficar(t(0, 0), t(1, 0))
comp_mat(3, i - xmin) = t(0, 0) - 273 ' t_chapa for Matte
black painting
comp_mat(7, i - xmin) = t(1, 0) - 273 ' t_vidrio for Matte
black painting
comp_mat(11, i - xmin) = t(2, 0) - 273 ' t_gap for Matte
black painting
comp_mat(15, i - xmin) = hvac          ' HVAC for Matte
black painting

Next

End Select

Return comp_mat
End Function

Private Function Cargas_Ilu_Heat_AC_para_graficar(ByVal xmax As Integer,
                                                    ByVal xmin As Integer) As
Double(,)
    Dim cargas(47, (xmax - xmin) + 1) As Double
    Dim ahorro(47, (xmax - xmin) + 1) As Double
    Dim cargas_y_ahorro(95, (xmax - xmin) + 1) As Double
    Dim provisional(5) As String
    Dim materiales(3) As String
    Dim ciudad_separacion(3) As String
    materiales = {"Aluminium anodized", "Clean metallic galvanized sheet",
"Matte black painting", "Glossy white painting"}
    ciudad_separacion = {"Bilbao5", "Bilbao30", "Sevilla5", "Sevilla30"}

    For i = xmin To xmax
        p = i / 100

```



```

' Bilbao 5 cm
' Las tres primeras lineas corresponden a Bilbao 5 Aluminium
anodized,
' las tres siguientes a Bilbao 5 Clean metallic galvanized sheet y
así sucesivamente.
' La línea 0 es ilu, la 1 es heat y la 2 es AC
For j = 0 To 3
  provisional = Calcular_Ilu_Heat_AC("Bilbao", materiales(j), p, 5)
  cargas(3 * j + 0, i - xmin) = CDb1(provisional(0))
  cargas(3 * j + 1, i - xmin) = CDb1(provisional(1))
  cargas(3 * j + 2, i - xmin) = CDb1(provisional(2))
  ahorro(3 * j + 0, i - xmin) = CDb1(provisional(3))
  ahorro(3 * j + 1, i - xmin) = CDb1(provisional(4))
  ahorro(3 * j + 2, i - xmin) = CDb1(provisional(5))
Next

' Bilbao 30 cm
For j = 0 To 3
  provisional = Calcular_Ilu_Heat_AC("Bilbao", materiales(j), p,
30)
  cargas(3 * (j + 4) + 0, i - xmin) = CDb1(provisional(0))
  cargas(3 * (j + 4) + 1, i - xmin) = CDb1(provisional(1))
  cargas(3 * (j + 4) + 2, i - xmin) = CDb1(provisional(2))
  ahorro(3 * (j + 4) + 0, i - xmin) = CDb1(provisional(3))
  ahorro(3 * (j + 4) + 1, i - xmin) = CDb1(provisional(4))
  ahorro(3 * (j + 4) + 2, i - xmin) = CDb1(provisional(5))
Next

' Sevilla 5 cm
For j = 0 To 3
  provisional = Calcular_Ilu_Heat_AC("Sevilla", materiales(j), p,
5)
  cargas(3 * (j + 8) + 0, i - xmin) = CDb1(provisional(0))
  cargas(3 * (j + 8) + 1, i - xmin) = CDb1(provisional(1))
  cargas(3 * (j + 8) + 2, i - xmin) = CDb1(provisional(2))
  ahorro(3 * (j + 8) + 0, i - xmin) = CDb1(provisional(3))
  ahorro(3 * (j + 8) + 1, i - xmin) = CDb1(provisional(4))
  ahorro(3 * (j + 8) + 2, i - xmin) = CDb1(provisional(5))
Next

' Sevilla 30 cm
For j = 0 To 3
  provisional = Calcular_Ilu_Heat_AC("Sevilla", materiales(j), p,
30)
  cargas(3 * (j + 12) + 0, i - xmin) = CDb1(provisional(0))
  cargas(3 * (j + 12) + 1, i - xmin) = CDb1(provisional(1))
  cargas(3 * (j + 12) + 2, i - xmin) = CDb1(provisional(2))
  ahorro(3 * (j + 12) + 0, i - xmin) = CDb1(provisional(3))
  ahorro(3 * (j + 12) + 1, i - xmin) = CDb1(provisional(4))
  ahorro(3 * (j + 12) + 2, i - xmin) = CDb1(provisional(5))
Next

Next

```

```

For i = 0 To 47
  For j = 0 To (xmax - xmin) + 1
    cargas_y_ahorro(i, j) = cargas(i, j)
    cargas_y_ahorro(i + 48, j) = ahorro(i, j)
  Next
Next

Return cargas_y_ahorro
End Function

Public Sub Grafico_materiales(ByVal vble_x As String,
                              ByVal vble_y As String,
                              ByVal titulo_grafico As String,
                              ByVal xmax As Integer,
                              ByVal xmin As Integer)

  Dim interval As Integer = FormPrincipal.numIntervalo.Value
  Dim comp_mat(15, (xmax - xmin) + 1) As Double
  Dim cargas_y_ahorro(95, (xmax - xmin) + 1) As Double
  Dim vbles_1(3) As String
  Dim vbles_2(3) As String
  vbles_1 = {"t_chapa", "t_vidrio", "t_gap", "HVAC"}
  vbles_2 = {"Bilbao5", "Bilbao30", "Sevilla5", "Sevilla30"}

  FormGraficos.btnTchapa.Visible = True
  FormGraficos.btnTvidrio.Visible = True
  FormGraficos.btnTgap.Visible = True
  FormGraficos.btnHVAC.Visible = True

  If (vble_x = "perf_ratio") Then
    FormGraficos.cboBilbao5.Visible = True
    FormGraficos.cboBilbao30.Visible = True
    FormGraficos.cboSevilla5.Visible = True
    FormGraficos.cboSevilla30.Visible = True
    FormGraficos.lblBilbao5.Visible = True
    FormGraficos.lblBilbao30.Visible = True
    FormGraficos.lblSevilla5.Visible = True
    FormGraficos.lblSevilla30.Visible = True
  End If

  FormGraficos.btnTchapa.BringToFront()
  FormGraficos.btnTvidrio.BringToFront()
  FormGraficos.btnTgap.BringToFront()
  FormGraficos.btnHVAC.BringToFront()
  FormGraficos.cboBilbao5.BringToFront()
  FormGraficos.cboBilbao30.BringToFront()
  FormGraficos.cboSevilla5.BringToFront()
  FormGraficos.cboSevilla30.BringToFront()

  If Incluye_nombre(vble_y, vbles_1) Then
    FormGraficos.chartCompararMateriales.Visible = True
    FormGraficos.chartConsumoAhorro.Visible = False
    comp_mat = Valores_grafico_materiales(vble_x, xmin, xmax)
  
```

```

Else
  FormGraficos.chartConsumoAhorro.Visible = True
  FormGraficos.chartCompararMateriales.Visible = False
  cargas_y_ahorro = Cargas_Ilu_Heat_AC_para_graficar(xmax, xmin)
End If

' Creo el gráfico
FormGraficos.chartConsumoAhorro.ChartAreas(0).AxisX.Title = vble_x
FormGraficos.chartConsumoAhorro.ChartAreas(0).AxisX.TitleFont = New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)
FormGraficos.chartConsumoAhorro.ChartAreas(0).AxisX.Interval =
Round((xmax - xmin) / 10, 0)
FormGraficos.chartConsumoAhorro.ChartAreas(0).AxisY.Title = vble_y
FormGraficos.chartConsumoAhorro.ChartAreas(0).AxisY.TitleFont = New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)
FormGraficos.chartConsumoAhorro.ChartAreas(1).AxisX.Title = vble_x
FormGraficos.chartConsumoAhorro.ChartAreas(1).AxisX.TitleFont = New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)
FormGraficos.chartConsumoAhorro.ChartAreas(1).AxisX.Interval =
Round((xmax - xmin) / 10, 0)
FormGraficos.chartConsumoAhorro.ChartAreas(1).AxisY.Title = vble_y
FormGraficos.chartConsumoAhorro.ChartAreas(1).AxisY.TitleFont = New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)

' Calculo el máximo valor del eje y si se grafican temperaturas para
mantener una escala fija en tiempo
' de ejecución.

Dim maximum As Double = -10000
Dim minimum As Double = 10000
Dim minHVAC As Double = 100000
For i = 0 To 11
  If comp_mat(i, (xmax - xmin)) > maximum Then
    maximum = comp_mat(i, (xmax - xmin))
  End If

  If comp_mat(i, 0) < minimum Then
    minimum = comp_mat(i, 0)
  End If

  If (i > 7) And comp_mat(i + 4, (xmax - xmin)) < minHVAC Then
    minHVAC = comp_mat(i + 4, (xmax - xmin))
  End If
Next

maximum = maximum + 5
minimum = minimum - 5

For i = 0 To 3
  FormGraficos.chartCompararMateriales.Series(i).Points.Clear()
  FormGraficos.chartConsumoAhorro.Series(i).Points.Clear()
  FormGraficos.chartConsumoAhorro.Series(i + 4).Points.Clear()

  If vble_y = "HVAC" Then

```

```

        FormGraficos.chartCompararMateriales.Series(i).ChartType      =
DataVisualization.Charting.SeriesChartType.Column
        FormGraficos.chartCompararMateriales.Series(i).CustomProperties =
"PointWidth=0.5"
        Else
        FormGraficos.chartCompararMateriales.Series(i).ChartType      =
DataVisualization.Charting.SeriesChartType.Spline
        End If
    Next

    Select Case vble_y
        Case "t_chapa"
            For j = 0 To (xmax - xmin)

FormGraficos.chartCompararMateriales.Series(0).Points.AddXY(xmin + j, comp_mat(0,
j))

FormGraficos.chartCompararMateriales.Series(1).Points.AddXY(xmin + j, comp_mat(1,
j))

FormGraficos.chartCompararMateriales.Series(2).Points.AddXY(xmin + j, comp_mat(2,
j))

FormGraficos.chartCompararMateriales.Series(3).Points.AddXY(xmin + j, comp_mat(3,
j))

                Next

                FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Maximum
= Round(maximum + 5, 0)
                FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Minimum
= Round(minimum - 5, 0)
                FormGraficos.chartCompararMateriales.ChartAreas(0).AxisX.Title =
vble_x

FormGraficos.chartCompararMateriales.ChartAreas(0).AxisX.TitleFont      =      New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)
                FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Title =
vble_y

FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.TitleFont      =      New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)

        Case "t_vidrio"
            For j = 0 To (xmax - xmin)

FormGraficos.chartCompararMateriales.Series(0).Points.AddXY(xmin + j, comp_mat(4,
j))

FormGraficos.chartCompararMateriales.Series(1).Points.AddXY(xmin + j, comp_mat(5,
j))

FormGraficos.chartCompararMateriales.Series(2).Points.AddXY(xmin + j, comp_mat(6,
j))

```

```

FormGraficos.chartCompararMateriales.Series(3).Points.AddXY(xmin + j, comp_mat(7,
j))
    Next

    FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Maximum
= Round(maximum + 5, 0)
    FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Minimum
= Round(minimum - 5, 0)
    FormGraficos.chartCompararMateriales.ChartAreas(0).AxisX.Title =
vble_x

FormGraficos.chartCompararMateriales.ChartAreas(0).AxisX.TitleFont = New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)
    FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Title =
vble_y

FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.TitleFont = New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)

    Case "t_gap"
        For j = 0 To (xmax - xmin)

FormGraficos.chartCompararMateriales.Series(0).Points.AddXY(xmin + j, comp_mat(8,
j))

FormGraficos.chartCompararMateriales.Series(1).Points.AddXY(xmin + j, comp_mat(9,
j))

FormGraficos.chartCompararMateriales.Series(2).Points.AddXY(xmin + j,
comp_mat(10, j))

FormGraficos.chartCompararMateriales.Series(3).Points.AddXY(xmin + j,
comp_mat(11, j))
            Next

            FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Maximum
= Round(maximum + 5, 0)
            FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Minimum
= Round(minimum - 5, 0)
            FormGraficos.chartCompararMateriales.ChartAreas(0).AxisX.Title =
vble_x

FormGraficos.chartCompararMateriales.ChartAreas(0).AxisX.TitleFont = New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)
            FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Title =
vble_y

FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.TitleFont = New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)

        Case "HVAC"
            For j = 0 To (xmax - xmin) Step interval

```

```

FormGraficos.chartCompararMateriales.Series(0).Points.AddXY(xmin + j,
comp_mat(12, j))

FormGraficos.chartCompararMateriales.Series(1).Points.AddXY(xmin + j,
comp_mat(13, j))

FormGraficos.chartCompararMateriales.Series(2).Points.AddXY(xmin + j,
comp_mat(14, j))

FormGraficos.chartCompararMateriales.Series(3).Points.AddXY(xmin + j,
comp_mat(15, j))
    Next

    FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Maximum
= FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.MaximumAutoSize
    FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Minimum
= Round(minHVAC - 5, 0)
    FormGraficos.chartCompararMateriales.ChartAreas(0).AxisX.Title =
vble_x

FormGraficos.chartCompararMateriales.ChartAreas(0).AxisX.TitleFont = New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)
    FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.Title =
vble_y

FormGraficos.chartCompararMateriales.ChartAreas(0).AxisY.TitleFont = New
Font("Microsoft Sans Serif", 14, FontStyle.Bold, GraphicsUnit.Pixel)

    Case "Bilbao5_Ilu"
    For j = 0 To (xmax - xmin)
    FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(0, j))
    FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(3, j))
    FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(6, j))
    FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(9, j))
    FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(48, j))
    FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(51, j))
    FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(54, j))
    FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(57, j))
    Next

    Case "Bilbao5_Heat"
    For j = 0 To (xmax - xmin)
    FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(1, j))
  
```

```

        FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(4, j))
        FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(7, j))
        FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(10, j))
        FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(49, j))
        FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(52, j))
        FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(55, j))
        FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(58, j))
        Next

    Case "Bilbao5_AC"
        For j = 0 To (xmax - xmin)
            FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(2, j))
            FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(5, j))
            FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(8, j))
            FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(11, j))
            FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(50, j))
            FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(53, j))
            FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(56, j))
            FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(59, j))
        Next

    Case "Bilbao30_Ilu"
        For j = 0 To (xmax - xmin)
            FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(12, j))
            FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(15, j))
            FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(18, j))
            FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(21, j))
            FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(60, j))
            FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(63, j))
            FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(66, j))
            FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(69, j))

```

Next

```

Case "Bilbao30_Heat"
  For j = 0 To (xmax - xmin)
    FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(13, j))
    FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(16, j))
    FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(19, j))
    FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(22, j))
    FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(61, j))
    FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(64, j))
    FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(67, j))
    FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(70, j))
  Next

```

```

Case "Bilbao30_AC"
  For j = 0 To (xmax - xmin)
    FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(14, j))
    FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(17, j))
    FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(20, j))
    FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(23, j))
    FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(62, j))
    FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(65, j))
    FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(68, j))
    FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(71, j))
  Next

```

```

Case "Sevilla5_Ilu"
  For j = 0 To (xmax - xmin)
    FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(24, j))
    FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(27, j))
    FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(30, j))
    FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(33, j))
    FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(72, j))

```



```

        FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(75, j))
        FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(78, j))
        FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(81, j))
        Next

    Case "Sevilla5_Heat"
        For j = 0 To (xmax - xmin)
            FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(25, j))
            FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(28, j))
            FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(31, j))
            FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(34, j))
            FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(73, j))
            FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(76, j))
            FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(79, j))
            FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(82, j))
        Next

    Case "Sevilla5_AC"
        For j = 0 To (xmax - xmin)
            FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(26, j))
            FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(29, j))
            FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(32, j))
            FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(35, j))
            FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(74, j))
            FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(77, j))
            FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(80, j))
            FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(83, j))
        Next

    Case "Sevilla30_Ilu"
        For j = 0 To (xmax - xmin)
            FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(36, j))
            FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(39, j))

```

```

        FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(42, j))
        FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(45, j))
        FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(84, j))
        FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(87, j))
        FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(90, j))
        FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(93, j))
        Next

    Case "Sevilla30_Heat"
    For j = 0 To (xmax - xmin)
        FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(37, j))
        FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(40, j))
        FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(43, j))
        FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(46, j))
        FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(85, j))
        FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(88, j))
        FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(91, j))
        FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(94, j))
        Next

    Case "Sevilla30_AC"
    For j = 0 To (xmax - xmin)
        FormGraficos.chartConsumoAhorro.Series(0).Points.AddXY(xmin +
j, cargas_y_ahorro(38, j))
        FormGraficos.chartConsumoAhorro.Series(1).Points.AddXY(xmin +
j, cargas_y_ahorro(41, j))
        FormGraficos.chartConsumoAhorro.Series(2).Points.AddXY(xmin +
j, cargas_y_ahorro(44, j))
        FormGraficos.chartConsumoAhorro.Series(3).Points.AddXY(xmin +
j, cargas_y_ahorro(47, j))
        FormGraficos.chartConsumoAhorro.Series(4).Points.AddXY(xmin +
j, cargas_y_ahorro(86, j))
        FormGraficos.chartConsumoAhorro.Series(5).Points.AddXY(xmin +
j, cargas_y_ahorro(89, j))
        FormGraficos.chartConsumoAhorro.Series(6).Points.AddXY(xmin +
j, cargas_y_ahorro(92, j))
        FormGraficos.chartConsumoAhorro.Series(7).Points.AddXY(xmin +
j, cargas_y_ahorro(95, j))
        Next

```



BILBOKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE BILBAO

End Select

End Sub

End Module

modDatos

```
Imports System.IO
```

```
Module modDatos
```

```
Public Sub Asignar_datos_al_proyecto(nombre As String)
    ' Este procedimiento traduce los datos del resumen del proyecto a
    variables
```

```
    Dim proyecto As TreeNode
    Dim chapa As TreeNode
    Dim camara As TreeNode
    Dim vidrio As TreeNode
    Dim ubicacion As TreeNode
    Dim clima As TreeNode
    Dim resultados As TreeNode
    Dim nn As Integer
    nn = FormPrincipal.treeProyectos.Nodes.Count()

    For i = 0 To (nn - 1)
        If nombre = FormPrincipal.treeProyectos.Nodes(i).Text Then
            proyecto = FormPrincipal.treeProyectos.Nodes(i)
            chapa = proyecto.Nodes(0)
            camara = proyecto.Nodes(1)
            vidrio = proyecto.Nodes(2)
            ubicacion = proyecto.Nodes(3)
            clima = proyecto.Nodes(4)
            resultados = proyecto.Nodes(5)
            Exit For
        End If
    Next

    alpha_solar_chapa = Cdbl(Mid(chapa.Nodes(0).Nodes(1).Text, 19)) / 100
    epsilon_IR_chapa = Cdbl(Mid(chapa.Nodes(0).Nodes(2).Text, 17)) / 100
    p = Cdbl(Mid(chapa.Nodes(1).Text, 23)) / 100

    If Mid(camara.Nodes(0).Text, 18) = "5" Then
        FormPrincipal.opt5cm.Checked = True
    ElseIf Mid(camara.Nodes(0).Text, 18) = "30" Then
        FormPrincipal.opt30cm.Checked = True
    End If

    FormPrincipal.cboRegión.SelectedItem = Mid(ubicacion.Nodes(0).Text, 9)
    FormPrincipal.cboCity.SelectedItem = Mid(ubicacion.Nodes(1).Text, 9)
    FormPrincipal.cboFecha.SelectedItem = Mid(clima.Nodes(0).Text, 6)
    FormPrincipal.cboTipoDia.SelectedItem = Mid(clima.Nodes(1).Text, 14)
    Ir_sol = Cdbl(Mid(clima.Nodes(2).Text, 21))
    t_amb = Cdbl(Mid(clima.Nodes(3).Text, 28)) + 273

    Hacer_calculos()
    Posicionar_barras(Ir_sol, p, t_amb)
```

End Sub

Public Sub Rellenar_proyecto(nombre As String)

 If nombre = "" Then

 Exit Sub

 End If

 ' Este procedimiento hace lo contrario que el anterior

 Dim proyecto As TreeNode

 Dim chapa As TreeNode

 Dim camara As TreeNode

 Dim vidrio As TreeNode

 Dim ubicacion As TreeNode

 Dim clima As TreeNode

 Dim resultados As TreeNode

 Dim nn As Integer

 nn = FormPrincipal.treeProyectos.Nodes.Count()

 For i = 0 To (nn - 1)

 If nombre = FormPrincipal.treeProyectos.Nodes(i).Text Then

 proyecto = FormPrincipal.treeProyectos.Nodes(i)

 chapa = proyecto.Nodes(0)

 camara = proyecto.Nodes(1)

 vidrio = proyecto.Nodes(2)

 ubicacion = proyecto.Nodes(3)

 clima = proyecto.Nodes(4)

 resultados = proyecto.Nodes(5)

 Exit For

 End If

 Next

 chapa.Nodes(0).Nodes(0).Text = "Material: " & material

 chapa.Nodes(0).Nodes(1).Text = "Absortividad (%): " & CStr(alpha_solar_chapa * 100)

 chapa.Nodes(0).Nodes(2).Text = "Emisividad (%): " & CStr(epsilon_IR_chapa * 100)

 chapa.Nodes(1).Text = "Ratio perforación (%): " & CStr(p * 100)

 camara.Nodes(0).Text = "Separación (cm): " & separacion

 vidrio.Nodes(0).Text = "Absortividad (%): " & CStr(alpha_solar_vidrio * 100)

 vidrio.Nodes(1).Text = "Emisividad (%): " & CStr(epsilon_IR_vidrio * 100)

 vidrio.Nodes(2).Text = "Transmisividad_SW (%): " & CStr(tr_sw_vidrio * 100)

 vidrio.Nodes(3).Text = "Transmisividad_LW (%): " & CStr(tr_lw_vidrio * 100)

 ubicacion.Nodes(0).Text = "Región: " & FormPrincipal.cboRegión.Text

 ubicacion.Nodes(1).Text = "Ciudad: " & FormPrincipal.cboCity.Text

 clima.Nodes(0).Text = "Mes: " & FormPrincipal.cboFecha.Text

 clima.Nodes(1).Text = "Tipo de día: " & FormPrincipal.cboTipoDia.Text

 clima.Nodes(2).Text = "Irradiancia (W/m²): " & CStr(Ir_sol)

 clima.Nodes(3).Text = "Temperatura ambiente (°C): " & CStr(t_amb - 273)

 resultados.Nodes(0).Nodes(0).Text = FormPrincipal.lblTchapa.Text

 resultados.Nodes(0).Nodes(1).Text = FormPrincipal.lblTvidrio.Text

 resultados.Nodes(0).Nodes(2).Text = FormPrincipal.lblTgap.Text

```

resultados.Nodes(1).Text = "HVAC (W/m²): " & FormPrincipal.lblHVAC.Text

    If (FormPrincipal.cboCity.Text = "Bilbao") Or (FormPrincipal.cboCity.Text
= "Sevilla") Then
        resultados.Nodes(2).Nodes(0).Text = "Iluminación: " &
FormConsumoAhorro.lviConsumoAhorro.Items(1).SubItems(1).Text
        resultados.Nodes(2).Nodes(1).Text = "Calefacción: " &
FormConsumoAhorro.lviConsumoAhorro.Items(3).SubItems(1).Text
        resultados.Nodes(2).Nodes(2).Text = "Aire acondicionado: " &
FormConsumoAhorro.lviConsumoAhorro.Items(5).SubItems(1).Text
        resultados.Nodes(2).Nodes(3).Text = "TOTAL: " &
FormConsumoAhorro.lviConsumoAhorro.Items(7).SubItems(1).Text
        resultados.Nodes(3).Nodes(0).Text = "Iluminación: " &
FormConsumoAhorro.lviConsumoAhorro.Items(1).SubItems(2).Text
        resultados.Nodes(3).Nodes(1).Text = "Incremento calefacción: " &
FormConsumoAhorro.lviConsumoAhorro.Items(3).SubItems(2).Text
        resultados.Nodes(3).Nodes(2).Text = "Aire acondicionado: " &
FormConsumoAhorro.lviConsumoAhorro.Items(5).SubItems(2).Text
        resultados.Nodes(3).Nodes(3).Text = "TOTAL: " &
FormConsumoAhorro.lviConsumoAhorro.Items(7).SubItems(2).Text
    Else
        resultados.Nodes(2).Nodes(0).Text = "Iluminación: "
        resultados.Nodes(2).Nodes(1).Text = "Calefacción: "
        resultados.Nodes(2).Nodes(2).Text = "Aire acondicionado: "
        resultados.Nodes(2).Nodes(3).Text = "TOTAL: "
        resultados.Nodes(3).Nodes(0).Text = "Iluminación: "
        resultados.Nodes(3).Nodes(1).Text = "Incremento calefacción: "
        resultados.Nodes(3).Nodes(2).Text = "Aire acondicionado: "
        resultados.Nodes(3).Nodes(3).Text = "TOTAL: "
    End If

End Sub

Public Sub Generar_proyecto(nombre As String)
    If nombre = "" Then
        Exit Sub
    End If

    Dim nodos_proyecto(5) As String
    Dim nodos_chapa(1) As String
    Dim nodos_chapa_mat(2) As String
    Dim nodos_vidrio(3) As String
    Dim nodos_ubicacion(1) As String
    Dim nodos_clima(3) As String
    Dim nodos_resultados(3) As String
    Dim nodos_resultados_temp(2) As String
    Dim nodos_resultados_cons(3) As String
    Dim nodos_resultados_ahor(3) As String
    Dim nn As Integer
    Dim proyecto As TreeNode
    Dim chapa As TreeNode
    Dim camara As TreeNode
    Dim vidrio As TreeNode
    Dim ubicacion As TreeNode

```

```

Dim clima As TreeNode
Dim resultados As TreeNode
Dim material As TreeNode
Dim temperaturas As TreeNode
Dim consumo As TreeNode
Dim ahorro As TreeNode

nodos_proyecto = {"Chapa", "Cámara", "Vidrio", "Ubicación", "Clima",
"Resultados"}
nodos_chapa = {"Material", "Ratio perforación (%): "}
nodos_chapa_mat = {"Material: ", "Absortividad (%): ", "Emisividad (%):
"}
nodos_vidrio = {"Absortividad: ", "Emisividad: ", "Transmisividad_SW: ",
"Transmisividad_LW: "}
nodos_ubicacion = {"Región: ", "Ciudad: "}
nodos_clima = {"Mes: ", "Tipo de día: ", "Irradiancia solar (W/m²): ",
"Temperatura ambiente (°C): "}
nodos_resultados = {"Temperaturas", "HVAC (W/m²): ", "Consumo (kWh)",
"Ahorro (%)"}
nodos_resultados_temp = {"T_chapa: ", "T_vidrio: ", "T_cámara: "}
nodos_resultados_cons = {"Iluminación: ", "Calefacción: ", "Aire
acondicionado: ", "TOTAL: "}
nodos_resultados_ahor = {"Iluminación: ", "Incremento calefacción: ",
"Aire acondicionado: ", "TOTAL: "}

FormPrincipal.treeProyectos.Nodes.Add(nombre)
nn = FormPrincipal.treeProyectos.Nodes.Count
proyecto = FormPrincipal.treeProyectos.Nodes(nn - 1)

' Crear nodos principales
For i = 0 To 5
    proyecto.Nodes.Add(nodos_proyecto(i))
Next

' Crear nodos secundarios
chapa = proyecto.Nodes(0)
camara = proyecto.Nodes(1)
vidrio = proyecto.Nodes(2)
ubicacion = proyecto.Nodes(3)
clima = proyecto.Nodes(4)
resultados = proyecto.Nodes(5)

For i = 0 To 1
    chapa.Nodes.Add(nodos_chapa(i))
Next

For i = 0 To 3
    vidrio.Nodes.Add(nodos_vidrio(i))
    resultados.Nodes.Add(nodos_resultados(i))
Next

camara.Nodes.Add("Separación (cm): ")

For i = 0 To 1

```

```

        ubicacion.Nodes.Add(nodos_ubicacion(i))
    Next

    For i = 0 To 3
        clima.Nodes.Add(nodos_clima(i))
    Next

    ' Crear los subnodos de los nodos secundarios
    material = chapa.Nodes(0)
    temperaturas = resultados.Nodes(0)
    consumo = resultados.Nodes(2)
    ahorro = resultados.Nodes(3)

    For i = 0 To 2
        material.Nodes.Add(nodos_chapa_mat(i))
        temperaturas.Nodes.Add(nodos_resultados_temp(i))
    Next

    For i = 0 To 3
        consumo.Nodes.Add(nodos_resultados_cons(i))
        ahorro.Nodes.Add(nodos_resultados_ahor(i))
    Next

End Sub

Public Sub Guardar_proyecto()
    Dim path As String
    FormPrincipal.sfdGuardarProyecto.Filter = "Archivo Edificios Fachada
Doble Piel (*.aadm)|*.aadm"

    If FormPrincipal.sfdGuardarProyecto.ShowDialog() <> DialogResult.Cancel
Then
        path = FormPrincipal.sfdGuardarProyecto.FileName
        Dim objDatosProyecto As New StreamWriter(path, False)
        objDatosProyecto.WriteLine(CStr(alpha_solar_chapa))
        objDatosProyecto.WriteLine(CStr(epsilon_IR_chapa))
        objDatosProyecto.WriteLine(CStr(p))
        objDatosProyecto.WriteLine(separacion)
        objDatosProyecto.WriteLine(FormPrincipal.cboRegión.Text)
        objDatosProyecto.WriteLine(FormPrincipal.cboCity.Text)
        objDatosProyecto.WriteLine(FormPrincipal.cboFecha.Text)
        objDatosProyecto.WriteLine(FormPrincipal.cboTipoDia.Text)
        objDatosProyecto.WriteLine(CStr(Ir_sol))
        objDatosProyecto.WriteLine(CStr(t_amb))
        objDatosProyecto.WriteLine(FormPrincipal.txtNombreProyecto.Text)
        objDatosProyecto.Close()
    End If

End Sub

Public Sub Abrir_proyecto()
    Dim path As String
    FormPrincipal.ofdAbrirProyecto.Filter = "Arcivo Edificios Fachada Doble
Piel (*.aadm)|*.aadm"

```



```

If FormPrincipal.ofdAbrirProyecto.ShowDialog() <> DialogResult.Cancel
Then
    path = FormPrincipal.ofdAbrirProyecto.FileName
    Dim objDatosProyecto As New StreamReader(path)
    Dim strDatosExtraidos(10) As String

    For i = 0 To 10
        strDatosExtraidos(i) = objDatosProyecto.ReadLine()
    Next

    objDatosProyecto.Close()

    If IsNumeric(CDbl(strDatosExtraidos(0))) And
(CDbl(strDatosExtraidos(0)) >= 0) And (CDbl(strDatosExtraidos(0)) <= 1)) Then
        alpha_solar_chapa = CDbl(strDatosExtraidos(0))
    Else
        MessageBox.Show("No se reconoce el contenido de este fichero.",
            "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error)
        Exit Sub
    End If

    If IsNumeric(CDbl(strDatosExtraidos(1))) And
(CDbl(strDatosExtraidos(1)) >= 0) And (CDbl(strDatosExtraidos(1)) <= 1)) Then
        epsilon_IR_chapa = CDbl(strDatosExtraidos(1))
    Else
        MessageBox.Show("No se reconoce el contenido de este fichero.",
            "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error)
        Exit Sub
    End If

    If IsNumeric(CDbl(strDatosExtraidos(2))) And
(CDbl(strDatosExtraidos(2)) >= 0.1) And CDbl(strDatosExtraidos(2)) <= 0.7) Then
        p = CDbl(strDatosExtraidos(2))
    ElseIf IsNumeric(CDbl(strDatosExtraidos(2))) And
(CDbl(strDatosExtraidos(2)) < 0.1) Then
        FormPrincipal.checkSinChapa.Checked = True
    ElseIf IsNumeric(CDbl(strDatosExtraidos(2))) And
(CDbl(strDatosExtraidos(2)) > 1) Then
        FormPrincipal.checkChapaMaciza.Checked = True
    Else
        MessageBox.Show("No se reconoce el contenido de este fichero.",
            "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error)
        Exit Sub
    End If

    If strDatosExtraidos(3) = "5" Then
        FormPrincipal.opt5cm.Checked = True
    ElseIf strDatosExtraidos(3) = "30" Then
        FormPrincipal.opt30cm.Checked = True
    Else

```

```

    MessageBox.Show("No se reconoce el contenido de este fichero.",
                    "Error",
                    MessageBoxButtons.OK,
    MessageBoxIcon.Error)
    Exit Sub
  End If

  Dim strRegiones(9) As String
  strRegiones = {"Australia", "China", "Ciudades Africanas", "Ciudades
de América Latina",
                "Ciudades Árabes", "Ciudades Asiáticas", "Ciudades
Europeas", "Ciudades Nórdicas",
                "EEUU", "España"}
  If Incluye_nombre(strDatosExtraidos(4), strRegiones) Then
    FormPrincipal.cboRegión.Text = strDatosExtraidos(4)
    FormPrincipal.cboRegión.SelectedItem = strDatosExtraidos(4)
  ElseIf strDatosExtraidos(4) = "Seleccione una región" Then
    FormPrincipal.cboRegión.Text = strDatosExtraidos(4)
  Else
    MessageBox.Show("No se reconoce el contenido de este fichero.",
                    "Error",
                    MessageBoxButtons.OK,
    MessageBoxIcon.Error)
    Exit Sub
  End If

  Dim strCiudades(25) As String
  strCiudades = {"Sidney", "Melbourne", "Shangai", "Beijing", "Hong
Kong", "Guangzhou", "Macau", "Cape Town",
                "Johannesburgo", "Douala", "Nairobi", "Bogotá", "Rio
de Janeiro", "Dubai", "Bangkok", "Nueva Delhi",
                "Bombay", "París", "Londres", "Estocolmo", "Nueva
York", "Los Ángeles", "Bilbao", "Madrid", "Sevilla",
                "Valencia"}
  If Incluye_nombre(strDatosExtraidos(5), strCiudades) Then
    FormPrincipal.cboCity.Text = strDatosExtraidos(5)
    FormPrincipal.cboCity.SelectedItem = strDatosExtraidos(5)
  ElseIf strDatosExtraidos(5) = "Seleccione una ciudad" Then
    FormPrincipal.cboCity.Text = strDatosExtraidos(5)
  Else
    MessageBox.Show("No se reconoce el contenido de este fichero.",
                    "Error",
                    MessageBoxButtons.OK,
    MessageBoxIcon.Error)
    Exit Sub
  End If

  Dim strMeses(11) As String
  strMeses = {"Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio",
"Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"}
  If Incluye_nombre(strDatosExtraidos(6), strMeses) Then
    FormPrincipal.cboFecha.Text = strDatosExtraidos(6)
    FormPrincipal.cboFecha.SelectedItem = strDatosExtraidos(6)
  ElseIf strDatosExtraidos(6) = "Seleccione un mes" Then
    FormPrincipal.cboFecha.Text = strDatosExtraidos(6)
  Else
    MessageBox.Show("No se reconoce el contenido de este fichero.",

```

```

                                "Error",                                MessageBoxButtons.OK,
MessageBoxIcon.Error)
    Exit Sub
End If

Dim strTipoDia(2) As String
strTipoDia = {"Soleado", "Parcialmente nublado", "Nublado"}
If Incluye_nombre(strDatosExtraidos(7), strTipoDia) Then
    FormPrincipal.cboTipoDia.Text = strDatosExtraidos(7)
    FormPrincipal.cboTipoDia.SelectedItem = strDatosExtraidos(7)
ElseIf strDatosExtraidos(7) = "Seleccione el tipo de día" Then
    FormPrincipal.cboTipoDia.Text = strDatosExtraidos(7)
Else
    MessageBox.Show("No se reconoce el contenido de este fichero.",
                                "Error",                                MessageBoxButtons.OK,
MessageBoxIcon.Error)
    Exit Sub
End If

If IsNumeric(CDbl(strDatosExtraidos(8))) And
(CDbl(strDatosExtraidos(8)) >= 0) And (CDbl(strDatosExtraidos(8)) <= 900) Then
    Ir_sol = CDbl(strDatosExtraidos(8))
Else
    MessageBox.Show("No se reconoce el contenido de este fichero.",
                                "Error",                                MessageBoxButtons.OK,
MessageBoxIcon.Error)
    Exit Sub
End If

If IsNumeric(CDbl(strDatosExtraidos(9))) And
(CDbl(strDatosExtraidos(9)) >= 223) And (CDbl(strDatosExtraidos(9)) <= 343) Then
    t_amb = CDbl(strDatosExtraidos(9))
Else
    MessageBox.Show("No se reconoce el contenido de este fichero.",
                                "Error",                                MessageBoxButtons.OK,
MessageBoxIcon.Error)
    Exit Sub
End If

FormPrincipal.txtNombreProyecto.Text = strDatosExtraidos(10)
Generar_proyecto(strDatosExtraidos(10))

Hacer_calculos()
Posicionar_barras(Ir_sol, p, t_amb)
Rellenar_proyecto(strDatosExtraidos(10))

End If
End Sub

End Module

```

ANEXO II: EJEMPLO DE INFORME. Caso *Bilbao_verano*.

BILBAO VERANO

PARÁMETROS DE CÁLCULO

Aspectos constructivos de la fachada:

CHAPA	Material	Tipo	<i>Aluminium anodized</i>
		Absortividad	50 %
	Emisividad	84 %	
	Ratio de perforación	50 %	
CÁMARA	Separación	30 cm	

Ubicación y clima:

UBICACIÓN	Región	<i>España</i>
	Ciudad	<i>Bilbao</i>
CLIMA	Mes	<i>Julio</i>
	Irradiancia	228,75 W/m ²
	Temperatura ambiente	20,4 °C

RESULTADOS

Temperaturas en la fachada y HVAC:

T_chapa	T_vidrio	T_cámara	HVAC
26,085 °C	22,412 °C	22,324 °C	-103,184 W/m ²

Consumo y ahorro anual:

	Iluminación	Calefacción	Aire acondicionado	TOTAL
Consumo	933,581 kWh	2569,019 kWh	2064,096 kWh	5566,696 kWh
Ahorro	-10,39 %	140,16 %	49,51 %	17,73 %

BIBLIOGRAFÍA

- [1] Á. Vidal Torreira, *Modelado del comportamiento térmico y fluido-dinámico de una fachada ventilada de chapa perforada*, Bilbao, 2016.
- [2] M. Shameri, M. Alghoul, K. Sopian, M. F. M.Zain y O. Elayeb, «Perspectives of double skin façade systems in buildings and energy saving,» *Renewable and Sustainable Energy Reviews*, nº 15, pp. 1468-1475, 2011.
- [3] E. Gratia y A. De Herde, «Natural cooling strategies efficiency in an office building with a double-skin façade,» *Energy and Buildings*, nº 36, pp. 1139-1152, 2004.
- [4] J. M. Blanco, P. Arriaga, E. Rojí y J. Cuadrado, «Investigating the thermal behavior of double-skin perforated sheet façades. Part A: Model characterization and validation procedure,» *Building and Environment*, nº 82, pp. 50-62, 2014.
- [5] Ç. Yunus A. y G. Afshin J., *Transferencia de calor y masa. Fundamentos y aplicaciones.*, Santa Fe: McGraw Hill, 2011.
- [6] J. M. Blanco, A. Buruaga, E. Rojí, J. Cuadrado y B. Pelaz, «Energy assessment and optimization of perforated metal sheet double skin façades through Design Builder. A case study in Spain.,» *Energy and Buildings*, nº 111, pp. 326-336, 2016.
- [7] *Directiva 2002/91/CE del Parlamento Europeo y del Consejo relativa a la eficiencia energética de los edificios*, 2002.
- [8] BOE, *Real Decreto 1027/2007*, 2007.
- [9] J. Foxall, *Visual Basic 2010. Paso a paso.*, Madrid: Anaya, 2010.
- [10] «www.lenguajesdeprogramacion.net,» [En línea]. Available: <https://lenguajesdeprogramacion.net/visual-basic/>. [Último acceso: 12 06 2018].
- [11] Microsoft®, «www.msn.com,» [En línea]. Available: <https://www.msn.com/es-cl/noticias/microsoftstore/%C2%BFqu%C3%A9-es-y-para-qu%C3%A9-sirve-visual-studio-2017/ar-AAAnLZL9>. [Último acceso: 12 06 2018].
- [12] W. H. Press, S. A. Teukolsky, W. T. Vetterling y B. P. Flannery, *Numerical Recipes. The art of scientific computing.*, Cambridge: Cambridge University Press, 2007.
- [13] G. Baldinelli, «Double skin façades for warm climate regions. Analysis of a solution with an integrated movable shading system,» *Building and Environment*, nº 44, pp. 1107-1118, 2009.