

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

MONITORIZACIÓN DEL CONSUMO ELÉCTRICO DE UN HOGAR: VISUALIZACIÓN VÍA WEB



Grado en Ingeniería
en Tecnologías Industriales

Trabajo Fin de Grado

Estudiante: Nahia Barriola Hernandorena

Director: Javier Marcos Álvarez

Codirector: Jesús López Taberna

Pamplona, 30 de Junio del 2015



AGRADECIMIENTOS

Antes que nada agradecer a mi tutor Javier Marcos Álvarez por su ayuda en los momentos en los que parecía que el proyecto no iba a salir adelante. Expresar también mi gratitud a Alberto Berrueta por las horas que ha pasado con mis compañeros y conmigo en el laboratorio y a Jesús López Taberna por sus consejos al inicio del proyecto.

Tampoco me puedo olvidar de los compañeros del laboratorio, familiares y amigos que me han aguantado durante estos años en la universidad.

Gracias a todos.

ABSTRACT

A smart counter allows the user to know the electricity consumption of their home at any moment through the representation of measured energy consumption data in different graphs. With this gadget, the consumer is able to increase the energetic efficiency through little habit changes.

This project depicts all the steps needed to make a web page which displays the electricity consumption of a house in different time horizons. Moreover, the most significant data regarding consumption, the electrical bill and some advice are shown.

To do that, the current data measured with a RaspberryPi and Arduino have been gathered in a data storage cloud called Google Drive. Afterwards, these data have been treated with Query Language Reference (Version 0.7) in order to turn current values into power and energy, creating data. Once the required work files have been obtained, an easy-to-understand web page has been programmed using R programming language and Shiny package. In the web page, designed with the CSS styles, only interesting parameters are shown.

KEY WORDS

- Smart counter
- Web page
- R-Studio
- Google Drive
- Energy efficiency

LABURPENA

Kontagailu adimendu edo inteligenteak erabiltzaileari bere kontsumoaren berri ematen dio uneoro, neurtutako datuen errepresentazio grafikoaren bidez. Honela, ohitura aldaketa txikien laguntzaz, kontsumitzailearen energia eraginkortasuna nabarmenki areagotzen da.

Lan honetan etxebizitza bateko kontsumo elektrikoari buruzko grafiko ezberdinak irudikatzen dituen Web orri bat egiteko beharrezkoak diren pausu guztiak azaltzen dira. Horrez gain, kontsumoari buruzko datu adierazgarrienak, faktura elektrikoa eta energia aurrezteko baliagarriak diren hainbat aholku ere bilduko ditu.

Horretarako, Google Drive izeneko biltegitratze hodeian bilduriko datuekin egin da lan. Hodei honetan, RaspberryPi eta Arduino batek neurturiko eta kalkulaturiko datuak gordetzen dira. Lana errazteko, balio hauek *Query Language Reference (0.7 bertsioa)*-ekin landu dira, horrela, korrante datuak abiapuntutzat hartuta, potentzia eta energia datuak biltzen dituzten taulak sortuz. Honela, lan egiteko beharrezko artxiboak lortzen dira eta gai gara RStudio izeneko programazio hizkeraren bidez eta Shiny liburutegiaren bidez, esandako web orria sortzeko. Lortzen den orria ulergarria da elektrizitatearen arloan jakinduria gutxi duen edozein pertsonarentzat, baliagarriak diren datuak soilik adierazten baitira. Web gunearen diseinua egiteko CSS estiloez ere baliatu gara.

HITZ GAKOAK

- Kontagailu adimendua
- Web orria
- R-Studio
- Google Drive
- Energia eraginkortasuna

RESUMEN

Un contador inteligente permite al usuario conocer en cualquier instante el consumo de electricidad del hogar mediante la representación de los datos medidos en diferentes gráficas. Esto permite al consumidor incrementar notoriamente la eficiencia energética mediante pequeños cambios de hábitos.

Este proyecto muestra todos los pasos necesarios a seguir para construir una página Web empezando desde cero, para la visualización del consumo eléctrico de un hogar en diferentes horizontes temporales. Además, también se recogerán los datos más significativos de consumo, la factura eléctrica y una amplia variedad de consejos para el ahorro energético.

Para ello, se han recogido los datos de corriente sensados y calculados por una placa RaspberryPi y un Arduino en una nube de almacenamiento de datos denominada *Google Drive*. Después, estos se han tratado con *Query Language Reference (Version 0.7)* para pasar los valores de corriente a potencia y a energía, generando así tablas de datos. Una vez obtenidos los archivos de trabajo requeridos, con el lenguaje de programación RStudio y el paquete Shiny, se ha realizado la programación de un sitio web de fácil comprensión para un usuario que no está especializado en el ámbito de la electricidad, mostrando solamente los parámetros que le puedan interesar. Para el diseño de la página, se ha hecho uso de estilos CSS.

PALABRAS CLAVE

- Contador inteligente
- Página Web
- R-Studio
- Google Drive
- Eficiencia energética

ÍNDICE

1. INTRODUCCIÓN	8
2. OBJETIVOS	9
3. ESTADO DEL ARTE	10
4. ORGANIZACIÓN DEL PROYECTO	13
4.1 Partes implicadas	13
4.2 Decisiones iniciales	14
5. CONDICIONES DE PARTIDA.....	16
5.1 Planteamiento de requisitos.....	16
5.2 Datos de partida: Nube de almacenamiento.....	17
6. PÁGINA WEB.....	21
6.1 CREACIÓN DE TABLAS PARA LA PÁGINA WEB	21
6.1.1 Datos de partida: Google Drive	21
6.1.2 Elaboración de tablas: Query Language Reference	22
6.2 DISEÑO DE LA PÁGINA WEB	28
6.2.1 Introducción: R, RStudio y Shiny	28
6.2.2 Acceso a los datos de las URL.....	31
6.2.3 Aspecto de la página web	34
6.2.4 Estilo de la fuente: Archivo CSS.....	38
6.3 CONTENIDO DE LA WEB.....	39
6.3.1 Consumo diario, mensual y anual	40
6.3.2 Datos significativos.....	46
6.3.3 Factura eléctrica	49
6.3.4 Consejos	54
7. LA APLICACIÓN EN LA RED.....	59
8. VISUALIZACIÓN DEL CONSUMO ELÉCTRICO DE UN HOGAR.....	61
9. CONCLUSIONES.....	64
10. LÍNEAS FUTURAS.....	66
11. BIBLIOGRAFÍA	67
ANEXO 1: ARCHIVO readGoogleSheet	69
ANEXO 2: ARCHIVO cleanGoogleTable	70

ANEXO 3: ARCHIVO CSS	72
ANEXO 4: SCRIPT ui.R.....	73
ANEXO 5 : SRIPT server.R.....	90

ÍNDICE FIGURAS

Figura 1: Contador CM160 OWL.....	10
Figura 2: Efergy E2	10
Figura 3: Contador Envir Current Cost	11
Figura 4: Ejemplo del diseño de una página web para la visualización del consumo de energía.....	12
Figura 5: Aspecto de la RaspberryPi sin caja	13
Figura 6: Aspecto del Arduino sin caja negra	14
Figura 7: Formato de datos en la hoja de cálculo de Google Drive	21
Figura 8: Ejemplo de cómo obtener el código adecuado para añadirlo a nuestra URL. 24	
Figura 9: Tabla de energía horaria con Query Language reference	24
Figura 10: Tabla de energía diaria con Query Language Reference.....	25
Figura 11: Aspecto de RStudio.....	29
Figura 12: Aspecto del script ui.R	30
Figura 13: Aspecto del script server.R	30
Figura 14: Aspecto de la página web.....	31
Figura 15: Ejemplo de lo que ocurriría al aplicar la función cleanGoogleTable	32
Figura 16: Ejemplo de código para cargar datos de energía diaria	33
Figura 17: Esquema de los apartados de la página web.	35
Figura 18: Código necesario en ui.R para crear las pestañas.	37
Figura 19: Aspecto de la página web.....	37
Figura 20: Código para calcular el número de horas, días, meses y años.....	39
Figura 21: Acceso a la tabla	41
Figura 22: Guardar la salida de la información temporal.....	41
Figura 23: Ejemplo sobre la limitación de la salida. Datos del mes del año anterior. ...	42
Figura 24: Gráfica comparativa del consumo de energía del día actual con el día anterior.....	43
Figura 25: Aspecto de la pestaña de datos diarios.....	44
Figura 26: Aspecto de la pestaña del consumo mensual	45
Figura 27: Aspecto de la pestaña consumo anual	46
Figura 28: Aspecto de la pestaña del mes de máximo consumo	49
Figura 29: Aspecto de la pestaña de la factura eléctrica.....	53
Figura 30: Aspecto de la pestaña de consejos.....	58

Figura 31: Subida de la aplicación a la red 59

Figura 32: Ventana de los script necesarios para la implementación de la página web 59

Figura 33: Aspecto de la página web en la red..... 60

Figura 34: Aspecto de la pestaña consumo diario de la medidas en la microrred con el Arduino 62

Figura 35: Aspecto de la pestaña consumo mensual de las medidas en la microrred con el Arduino 62

Figura 36: Aspecto de la pestaña consumo anual de las medidas en la microrred con el Arduino 62

Figura 37: Error en la página web al fallar la Raspberry o el Arduino 63

ÍNDICE DE TABLAS

Tabla 1: Tabla comparativa entre Google Drive y One Drive [13]..... 20

Tabla 2: Funciones shiny para ui.R 36

Tabla 3: : Información de las pestañas de datos diarios, mensuales y anuales 40

Tabla 4:Tabla de frecuencias. Información sobre el número de días de cada año 45

Tabla 5: Tabla acumulativa días del año..... 47

Tabla 6: Tabla acumulativa días de los meses..... 47

Tabla 7: Cómo calcular los términos de la factura eléctrica 50

Tabla 8: Precio de potencia y energía en función del tipo de tarifas..... 51

1. INTRODUCCIÓN

Actualmente la electricidad consumida en casa es un bien que el usuario percibe como universal, ubicuo y poco relacionado con los gases de efecto invernadero. Los contadores de electricidad colocados en la mayoría de los hogares son para muchos usuarios difíciles de comprender, y en ciertos casos los consumidores de energía desconocen su ubicación, ya que en comunidades de vecinos se sitúan en espacios comunes para facilitar la lectura de los mismos a las compañías.

Para hacer frente a ese desconocimiento, y así impulsar al usuario a reducir el consumo, controlando el gasto innecesario y aumentando la eficiencia, aparecen nuevos contadores capaces de intercambiar información con distintos aparatos e incluso con Internet de forma instantánea. Estos son los llamados “Contadores inteligentes” e indican el consumo que se tiene en cualquier momento, permitiendo al consumidor visualizar esos datos y el ahorro que suponen pequeños cambios de hábitos.

Además, Internet se ha convertido en un medio de comunicación indispensable para diferentes ámbitos de la sociedad, teniendo especial relevancia en el ámbito académico o laboral. En la actualidad, cualquier persona puede acceder a la información de la que precise en cada instante mediante esta herramienta, como por ejemplo a las facturas de móvil, gas, electricidad u otras informaciones de gran importancia para un consumidor normal.

Por esto, mediante este Trabajo Fin de Grado se pretende crear una página web donde cualquier usuario que coloque el dispositivo que se va a diseñar en el contador de electricidad de su hogar, pueda acceder a los datos en tiempo real. Así, podrá tener los datos del consumo y de su factura, observando los instantes en los que el consumo se dispare y podrá intentar reducir esos picos.

Con el producto obtenido se garantiza una mayor rapidez, eficiencia y facilidad de la relación entre el proveedor de electricidad y los clientes. Este tipo de contadores, funcionan de una forma digital y permiten transferencias automáticas y complejas de información. Con la visualización de esta información en la página web, el cliente puede acceder a la información que desea conocer desde cualquier lugar donde tenga acceso a internet. De esta forma, el cliente puede conocer en todo momento cuánto y cómo está consumiendo.

2. OBJETIVOS

Este proyecto se lleva a cabo con otros tres proyectos en paralelo, los realizados por Joseba Revuelta [1], Víctor Erice[2] y Aritz Legarrea[3] . Mediante estos, se pretende realizar la monitorización del consumo eléctrico de un hogar. Los otros tres trabajos se encargan del sensado de la corriente y su posterior medida con un Arduino y con una placa Raspberry Pi respectivamente.

El objetivo conjunto de los cuatro proyectos es diseñar y construir un contador inteligente para la medida del consumo de electricidad de un hogar, para que así el consumidor pueda disponer de los datos de energía y de la factura desde cualquier punto del que disponga de acceso a la red. Además es importante que tenga un bajo coste, una fácil instalación y similares características a los contadores inteligentes comerciales, para su competitividad en el mercado.

Resulta interesante ampliar el posible uso del dispositivo a otros ámbitos como el de investigación, por ejemplo, para realizar comparaciones del consumo de energía de hogares de diferentes características.

Una vez medido el consumo eléctrico de un hogar y subidos los datos a un sistema de almacenamiento gratuito, con este Trabajo Fin de Grado, se pretende crear una página web para poder representar estos datos y que el cliente pueda disponer de ellos en cualquier lugar en el que disponga de acceso a internet.

Para la implementación de la página web se hace uso de Shiny, mediante la programación con Rstudio, que evita el uso de JavaScript y HTML. Se diseña una página web sencilla y clara, con diferentes pestañas para que el usuario fácilmente pueda escoger la información que desea conocer.

En esta página web se recogerán los datos de la forma más clara posible mediante gráficas, para que un cliente no especializado en el tema pueda interpretar los datos fácilmente. Además del consumo en diferentes espacios de tiempo, se pretende calcular una aproximación de la factura eléctrica que le llegará al cliente al final de cada mes. Esta información de la factura eléctrica, solo servirá para aquellos usuarios que dispongan de una tarifa sin discriminación horaria, en la que la electricidad tenga el mismo precio independientemente del instante del consumo.

La página web, además del objetivo informativo sobre el consumo realizado, tendrá el fin de concienciar, donde mediante diferentes consejos se pretenderá que el cliente disminuya su consumo eléctrico para que así tenga un ahorro económico, al mismo tiempo que ayude a cuidar el medioambiente.

3. ESTADO DEL ARTE

En un previo análisis sobre el mercado actual de los contadores inteligentes, se aprecia la distinción de tres alternativas. La primera de ellas es la más sencilla, ya que los datos medidos únicamente se pueden visualizar en una pantalla LCD. En este caso, el dispositivo es muy sencillo para el uso de personas no especializadas, debido a que en la pequeña pantalla se presentan gráficas de forma clara donde se visualiza el consumo de energía en diferentes horizontes de tiempo. Son ejemplo de ello el contador *CM160 de la marca OWL* [4] o *Efergy E2* de la marca Efergy [5].



Figura 1: Contador CM160 OWL

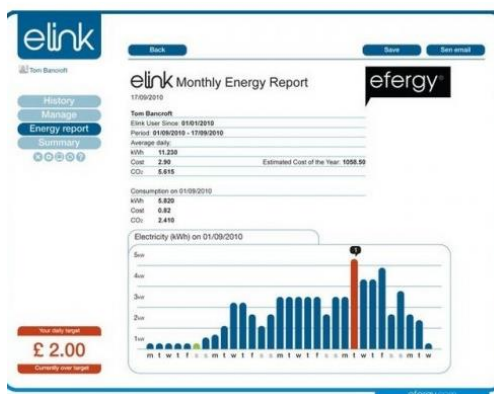


Figura 2: Efergy E2

Estas dos opciones permiten una rápida visualización del consumo de energía en diferentes intervalos de tiempo, así como una aproximación de la factura eléctrica.

Algo más sofisticada es la visualización de datos mediante una pantalla LCD y la exportación de datos a un ordenador mediante un USB. En este caso, además de tener

la simplicidad del caso anterior, presenta una pequeña mejora, ya que se puede tratar la información obtenida desde otro dispositivo y realizar gráficas, aunque los datos deban ser previamente descargados a un USB. Uno de los modelos más significativos encontrado en la red es el medidor de energía Envir, del fabricante Current Cost [6], que dispone de las características citadas.



Figura 3: Contador Envir Current Cost

En la figura 3 se puede observar que se presentan más detalles sobre el consumo, además de otras características meteorológicas, y que el aspecto que muestra en pantalla es más atractivo. Cabe destacar que en el caso de la figura 2, los datos se muestran en una pequeña pantalla LCD y en este caso en una pantalla del ordenador, por eso, la cantidad de datos es superior y la calidad de la imagen también.

La última opción que se va a presentar es la exportación de datos a una nube y su posterior visualización vía página web. Esta es la que más se aproxima a los requerimientos del proyecto, ya que se permite acceder a los datos de consumo desde cualquier punto con acceso a Internet, sin una previa descarga de los mismos.

Uno de los ejemplos más completos que se ha analizado es el contador de “engage.efergy”[7], donde se pueden ver los datos más relevantes del consumo en una página web. Además, se ofrece la posibilidad de descargar los datos que se recogen en la misma página web. Para alcanzar el objetivo de dar la información necesaria para el consumidor de una forma clara y sencilla, hace uso de diagramas de barras en los que se muestra la energía consumida a lo largo del día, de la semana, del mes o del año, además del consumo instantáneo de potencia, de las emisiones de CO₂ e información adicional de la climatología.



Figura 4: Ejemplo del diseño de una página web para la visualización del consumo de energía

4. ORGANIZACIÓN DEL PROYECTO

4.1 Partes implicadas

En la primera parte del proyecto realizada por Joseba Revuelta [1] se implementa el sensado de la corriente y su posterior volcado a una placa Arduino y otra RaspberryPi. Para ello, se debe crear un circuito eléctrico para adaptar la señal a la entrada de cada una de esas placas. Para el caso del Arduino, es necesario un divisor de tensión y un amplificador, debido a que la entrada al mismo es de un rango de entre 0 y 5V, y no se consiguen estos valores únicamente con el divisor de tensión. En cambio, para la entrada de la RaspberryPi hay que acondicionar la señal entre unos valores de 0 y 3.3V. En este caso, no es necesario el amplificador, ya que con el uso de un divisor de tensión adecuado, se obtienen valores de tensión que se acercan mucho al máximo valor de entrada permitido por la RaspberryPi. Por tanto, serán necesarios dos circuitos diferentes, uno para cada placa.

En la segunda parte del trabajo, hay dos subproyectos trabajando en paralelo. Por un lado la medida de la corriente mediante Arduino, que es realizada por Victor Erice[2], y por otro en una placa RaspberryPi, por Aritz Legarrea [3]. En el caso del primero de ellos, se necesitan dos Arduinos que trabajen a la vez, un Arduino Mini que se encarga de muestrear la señal aproximadamente cada un milisegundo y después de realizar una media cada 1.5 segundos, envía esa media mediante radiofrecuencia al Arduino Uno. Este último, realiza la media cada 10 minutos y sube ese dato a la red. El otro dispositivo empleado para la realización de la medida de corriente, es la RaspberryPi 2 modelo B, que muestrea la señal con la misma frecuencia, para realizar después cada 10 minutos mediante el método de los trapecios el cálculo de la corriente.

En esta segunda parte, hay tres objetivos fundamentales. Realizar la adecuada conexión entre los dispositivos y el sensado, la programación para el cálculo de la corriente cada 1ms y la media diezminutal, y por último, la subida de estos datos diezminutales de corriente a una nube de datos. En las siguientes imágenes se puede ver el aspecto que adoptan los dos contadores tras las dos primeras etapas. Cabe destacar, que no es el aspecto final, ya que se introducen en una caja negra.

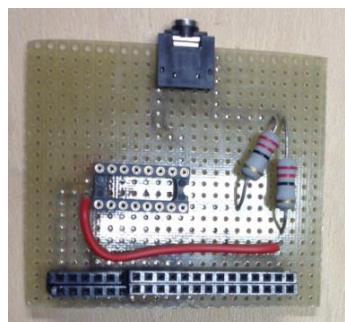


Figura 5: Aspecto de la RaspberryPi sin caja

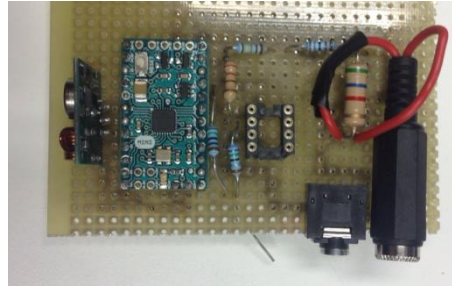


Figura 6: Aspecto del Arduino sin caja negra

En la presentación de este proyecto se parte de los datos guardados en una hoja de cálculo que se encuentra almacenada en google drive. En la primera columna de esta hoja, aparece la fecha en la que se realizan las medidas en el formato *"día/mes/año horas:minutos:segundos"*, y en la segunda columna las medidas de corriente. Cada diez minutos se sube un nuevo dato de fecha y corriente y se coloca debajo del último dato recogido hasta ese instante, por tanto, la hoja de cálculo se está refrescando todo el rato con los nuevos datos.

En esta última parte del proyecto, el objetivo final es la manipulación de estos datos para obtener información característica del consumo del hogar en el que se están realizando las medidas. Para una visualización y representación fácil y accesible desde cualquier lugar, la representación de estos datos se realiza mediante una página web.

4.2 Decisiones iniciales

Como ya se ha mencionado anteriormente, para comenzar con la realización del diseño de la página web, es necesario realizar las otras partes previamente. Para ello, todas las partes que constituyen el proyecto, deben ponerse de acuerdo para seguir la misma línea a la hora de trabajar.

Uno de los primeros aspectos a tratar es la frecuencia con la que se deben tomar los datos. Para ello, debido a que se tiene una onda a 50HZ, se cree conveniente tomar datos de la corriente cada 1m, para evitar el fenómeno aliasing. Una vez medidos estos datos, se realizarán las medias correspondientes a intervalos de tiempo de 10minutos para la posterior subida de esos datos a internet, ya que para nuestra aplicación es suficiente con tener 6 datos para cada hora.

Otra de las decisiones iniciales para la adecuada coordinación entre las partes es la elección de la forma de conexión a Internet, ya que aunque en un principio parezcan obvias las ventajas del wifi por ejemplo por la ausencia de cables, encarece mucho el contador diseñado con el Arduino. Por ello, se decide el diseño mediante dos Ardui-

nos en el que el primero de ellos manda la señal al segundo mediante radiofrecuencia, y este último sube los datos a Internet mediante la conexión al router con un cable Ethernet. La RaspberryPi no presenta ningún problema para subir los datos vía wifi, y por ello se opta por esa opción.

Se cree conveniente realizar únicamente el sensado de la corriente, ya que aunque también resultaría interesante el sensado de la tensión para así poder realizar el cálculo del factor de potencia y así obtener un dato más exacto de la potencia consumida, complicaría mucho el proyecto.

La decisión sobre el tipo de archivo a usar para el almacenamiento de datos se toma por los conocimientos previos en la materia, desechándose por completo la posibilidad del uso de una base de datos y optándose por una hoja de cálculo. La nube de datos escogida para este almacenamiento es Google Drive (a continuación se explica el porqué) y el formato de los datos a subir será el indicado en el apartado anterior.

Aunque el tipo de datos a mostrar en la web influya únicamente a la última parte del proyecto, es decir, el correspondiente a este Trabajo Final de Grado, entre todos los participantes del proyecto se tomó la decisión de mostrar por lo menos los datos de potencia instantánea, datos diarios, mensuales y anuales.

5. CONDICIONES DE PARTIDA

5.1 Planteamiento de requisitos

A lo largo de este Trabajo Fin de Grado se lleva a cabo el análisis, el diseño y la implementación de la página web “Conoce tu consumo”. El principal objetivo de esta página web es que el cliente pueda acceder a datos relevantes sobre su consumo de electricidad de una forma sencilla. Así, estos datos de interés para cualquier consumidor se pueden obtener desde cualquier dispositivo del que se tenga acceso a internet, y a su vez, se ayuda a una disminución del consumo de electricidad con su consiguiente reducción en la factura.

Para ello, es indispensable realizar el análisis para determinar los requisitos que debe cumplir para el cliente esta página web. Hay que tener en cuenta que esta debe ser sencilla, ya que su uso está dirigido a cualquier usuario que quiera disponer de un contador inteligente, y por tanto conocer el consumo que está teniendo en cada instante a través de la página web. Por todo esto, la plataforma virtual debe ser de fácil manejo, intuitiva y rápida.

Además, el uso masivo de los dispositivos electrónicos lleva a la necesidad de la creación de diferentes pestañas, para que en cada una de ellas se tenga la mínima información posible y se visualice rápidamente, sin tener que estar buscando en una larga página la información que se precise en cada instante.

“Conoce tu consumo”, es una página informativa que deberá cubrir las siguientes funciones:

- Diseñada para gente no cualificada en el ámbito de la electricidad.
- Intuitiva y de fácil interacción.
- Interfaz sencilla que incluya un logo y una imagen que venga acorde con el tema que se trata.
- Dispondrá de un menú horizontal con diferentes pestañas, con las distintas partes que tendrá la página: consumo diario de energía, consumo mensual de energía, consumo anual de energía, datos significativos, factura eléctrica y consejos para el ahorro de energía.
- En cada una de las pestañas se dan datos representativos y gráficas características para que el cliente pueda comprender fácilmente cuanto está consumiendo, y si ese valor es más alto de lo normal o no.
- Los datos se deben refrescar de una forma que los valores que el usuario vea en la página web sean realmente representativos. Debido a que las medidas se realizan cada 10 minutos, es conveniente que por lo

menos se refresque cada 5 minutos y así tener un error de como mucho 4min 59seg.

- En la pestaña de datos significativos habrá diferentes subpestañas: Mes del año de mayor consumo, mes del año de menor consumo, día del año de mayor consumo y día del año de menor consumo. En los casos del mes de mayor y menor consumo, además de visualizar los datos de energía, se añadirá el dato de la factura de ese mes, para que así el cliente vea la diferencia entre esos días o meses característicos.
- En la pestaña de la factura eléctrica, se dispondrá de dos subpestañas para poder ver la factura del mes anterior y del mes actual. Se intentará proporcionar al cliente de la forma más detallada y clara posible la información de la factura hasta el instante en el que se esté realizando la consulta.
- En la pestaña de consejos el usuario dispondrá de suficientes recomendaciones a seguir para una reducción del consumo que se notará en la factura eléctrica.

5.2 Datos de partida: Nube de almacenamiento

Desde el inicio del proyecto quedó claro que era necesario el almacenamiento de datos en la nube. Para ello se realiza el análisis de los mejores 5 servicios para el almacenamiento de archivos: Dropbpox, OneDrive, Box, Mega y GoogleDrive. Una de las claves de estos servicios es la privacidad de los mismos, ya que podemos elegir quién puede ver cada cosa.

Dropbox [8]

Este no es el primer almacenamiento en “la nube” que se creó, pero sí que es uno de los que mejor ha sabido adaptarse a las necesidades de los usuarios ofreciendo un completo acceso multiplataforma. La gestión y edición de los archivos personales se puede realizar desde cualquier dispositivo del usuario: smartphones, tablets y ordenadores. Así se evita llevarlos almacenados en las memorias de sus dispositivos.

Una de las claves del éxito de este servicio es su acertado interfaz, en el cual realiza la gestión de los archivos con vista en árbol. Otra de las características importantes de Dropbox es su sencillez, ya que se centra únicamente en ofrecer un espacio virtual, acercándose a un disco duro.

El espacio gratuito que ofrece esta nube de datos es de 2GB, pero es posible aumentar este espacio en 500MB adicionales por cada amigo que invite-mos y cree una cuenta, hasta un límite de 16GB. También es posible aumentar el espacio de almacenamiento mediante una forma de pago o por instalar dife-

rentes programas o “linkear” Dropbox en diferentes aplicaciones. El espacio total gratuito que se puede obtener es de 24GB.

OneDrive [9]

OneDrive es la propuesta de Microsoft para el almacenamiento de datos en la nube. Como en el anterior servicio de almacenamiento, en este también podemos guardar y compartir todo tipo de archivos gratuitamente hasta 15GB, y a partir de ese espacio se puede aumentar el espacio disponible mediante pago hasta 1TB.

La gran diferencia con Dropbox es que este espacio de almacenamiento no es tan simple, ya que además del clásico almacenamiento de archivos dispone de servicios adicionales como Office Web Apps, que es la versión para internet del paquete ofimático más utilizado en todo el mundo. Esto permite crear, editar y compartir documentos sin importar el tipo de dispositivo en el que trabajes.

Además de su impecable funcionamiento para usuarios de Windows 8, también existe una app para iPhone y Android.

Box [10]

Box es otro de los espacios de almacenamiento en la nube más importante, pero este está más bien dirigido al ámbito empresarial. Está diseñada para la seguridad y facilidad de uso, y mediante esta plataforma los empleados pueden trabajar de una forma segura con diferentes equipos clientes y socios, desde cualquier lugar y dispositivo. Es una plataforma interesante para la protección de documentos confidenciales.

La versión gratuita de Box ofrece 10GB de espacio y se puede acceder a los archivos almacenados desde el ordenador y de los dispositivos Android y iPhone, ya que existe una app para ello. Este espacio se puede aumentar con una forma de pago.

Mega [11]

Mega es la plataforma creada por la misma persona que creó Megapload. Eso hace que sea una opción bastante polémica. El espacio creado dispone de 50GB de almacenamiento gratuito, pero una de sus mayores pegas es su lentitud debido a la cantidad de cuentas abiertas que hay.

Se puede ampliar el almacenamiento del que se dispone hasta un máximo de 4TB con una forma de pago. Aunque al principio la única forma de acceso a esta plataforma era por ordenador, en estos momentos también podemos encontrar diferentes apps para tablets y smartphones.

Google Drive [12]

Tal y como podemos ver por el nombre, se trata de la propuesta del gigante Google. Esta nube de datos está compuesta por la suma del servicio ofimático online de Google, llamado Google Docs, con el añadido de almacenamiento de archivos. Al igual que en los otros servicios estudiados, Google Drive sincroniza de forma automática los contenidos almacenados para poder trabajar con ellos desde cualquier dispositivo accediendo a nuestra cuenta.

Este servicio dispone de aplicaciones para Windows y Mac, así como de apps para Chrome OS, el iOS de Apple y Android. Además de ofrecer toda la privacidad de la que disponen los otros espacios de almacenamiento, un aspecto innovador e interesante es que se puede compartir cualquier archivo que se encuentre en Google Drive sea cual sea su extensión, utilizando una herramienta para generar un enlace público para que cualquier persona pueda acceder al documento.

Drive ofrece 15GB de almacenamiento de Google gratis para comenzar y guardar, crear, editar y compartir todo tipo de archivos. Se puede invitar fácilmente a otros usuarios a ver los archivos además de descargarlos y trabajar en ellos sin necesidad de enviar archivos adjuntos por correo electrónico.

Después de analizar las diferentes posibilidades de almacenamiento, desde el primer momento quedan descartadas las opciones de box, ya que está dirigido a empresas, y mega, debido a los problemas por su lentitud. Por tanto las opciones a analizar serían Dropbox, OneDrive y Google Drive. De estas tres opciones el servicio en el que menos almacenamiento gratuito disponemos es Dropbox, por lo que a pesar de su sencillez, esta también queda descartada.

Las únicas opciones que quedan son OneDrive y Google Drive, las cuales son muy parecidas, ya que los dos disponen de un servicio ofimático online para poder trabajar y el usuario decide si comparte su contenido o no. En cambio hay ciertas diferencias entre los dos:

	<i>Google Drive</i>	<i>OneDrive</i>
<i>Almacenamiento</i>	15GB	15 GB
<i>Almacenamiento extra gratis</i>	-	15 GB (por instalarlo en Android o los) 500 MB (por referidos, máximo 10/5 GB)
<i>Total Gratis</i>	15GB	35GB

<i>Restricción del tamaño de archivo</i>	5TB	10GB
<i>Posibilidad de aumentar el espacio de almacenamiento de forma gratuita</i>	No	Sí
<i>Posibilidad de aumentar el espacio de almacenamiento mediante pago</i>	Sí. Hasta 30TB	Sí. Hasta 1TB
<i>Chat y trabajo simultáneo</i>	Puedes hablar con otras personas mediante un chat y trabajar simultáneamente.	No hay chat, aunque puede trabajarse simultáneamente con otras personas
<i>Mail asociado</i>	Gmail	hotmail

Tabla 1: Tabla comparativa entre Google Drive y One Drive [13]

En este proyecto, de todas las características comentadas nos interesa tener un documento de hoja de cálculo modificable para ir rellenando los datos medidos, por eso nos hemos centrado en estos dos servicios, que disponen de servicios online ofimáticos.

Entre estos dos servicios, aunque el que más almacenamiento gratuito tiene es OneDrive, el que mayor tamaño de archivos admite es GoogleDrive, y considerando que su capacidad de almacenamiento es suficiente, esta es la nube escogida. Cabe destacar, que en la elección también ha influido la facilidad de trabajar con google desde R, ya que existen diferentes funciones y paquetes que facilitan el trabajo.

6. PÁGINA WEB

6.1 CREACIÓN DE TABLAS PARA LA PÁGINA WEB

6.1.1 Datos de partida: Google Drive

Tal y como hemos dicho, al inicio de este trabajo, entre todos los participantes del proyecto se decidieron varios puntos a seguir: los datos se subirán a la nube de datos Google Drive, en dos columnas, donde en la primera se escribirá la fecha con el formato “*día/mes/año horas:minutos:segundos*” y en la segunda se situará la corriente en Amperios. Además estos datos se subirán cada diez minutos, teniendo así 6 datos para cada hora. Cabe destacar que a lo largo de la memoria se trabajará con una tabla de datos con medidas aleatorias de corriente de más de un año, para así poder ver el correcto funcionamiento de la página web. En la siguiente figura se muestra dicho formato y las casillas que se rellenarían al pasar 10 minutos.

	A	B	C	D
62619	11/03/2015 20:20	1,410869565		
62620	11/03/2015 20:30	0,972056084		
62621	11/03/2015 20:40	1,23188768		
62622	11/03/2015 20:50	1,484782609		
62623	11/03/2015 21:00	0,991883816		
62624	11/03/2015 21:10	1,069452106		
62625	11/03/2015 21:20	1,682608696		
62626	11/03/2015 21:30	1,083926714		
62627	11/03/2015 21:40	1,447707532		
62628	11/03/2015 21:50	0,580434783		
62629				

Figura 7: Formato de datos en la hoja de cálculo de Google Drive

Es necesario hacer totalmente público este archivo para más tarde poder acceder a él desde Rstudio. Para eso, primero clicaremos en *Archivo/ Publicar en la web/*

Publicar. Esto no es suficiente, ya que no se puede acceder a la página sin cuenta de Google Drive, por ello, es necesario ir a la página principal de Google Drive, clicar con el botón derecho en el archivo con el que se quiere trabajar, y elegir la opción compartir. Una vez hecho esto, el archivo es completamente público y cualquiera puede acceder al mismo. La mayor ventaja de esta característica es que no es necesario programar una página web donde se tenga que insertar el nombre de usuario y contraseña, lo que facilita bastante la programación. Muy unido a esto, se presenta su mayor desventaja, y es que cualquiera que tenga el link completo de la página de google donde se tienen los datos de consumo, puede acceder a esos datos.

6.1.2 Elaboración de tablas: Query Language Reference

En un primer momento se barajó la posibilidad de cargar el archivo público generado desde R cada diez minutos, para así realizar todos los cálculos pertinentes en R, pero más tarde se llegó a la conclusión de que era más práctico trabajar con tablas más pequeñas en las que ya está calculada la potencia o la energía, ya que una tabla tan grande hacía que el tiempo que le costaba refrescarse fuese muy alto.

Para la realización de unas tablas más pequeñas, en primer lugar se debe decidir cuáles van a ser los intervalos de tiempo que se van a analizar. Se cree conveniente el análisis de datos horarios, diarios, mensuales y anuales. Todos estos datos tienen sentido únicamente si se trabaja con datos de energía y no de potencia. Por otro lado, otro dato de interés para el usuario puede ser el último dato de potencia que se mide. En este último caso es muy importante poder acceder solamente a ese dato, ya que si cargáramos todo el archivo tendríamos 52560 datos por cada año (6datos x 24h x 365días), que se irían acumulando a lo largo de los años, y solamente necesitaríamos el último.

Al principio se pensó en la posibilidad de realizar los cálculos necesarios para conocer los datos interesantes en el mismo archivo de google drive en diferentes hojas del mismo libro, y así acceder a la hoja que nos interesara en cada momento. Viendo la complejidad de esta opción y lo costoso que sería en cuanto al tiempo empleado para realizar todas esas hojas, se analizó la posibilidad de trabajar con **Query Language Reference (Version 0.7)** [14].

Esta es una forma de acceder y manipular las tablas creadas en Google Drive trabajando de una forma muy similar al lenguaje SQL (Structured Query Language). Estos lenguajes están vinculados a la gestión de bases de datos y permiten la especificación de distintas operaciones entre los datos que se encuentran en este caso en la tabla. Gracias al uso del álgebra y de cálculos relacionales, mediante este tipo de len-

guajes se pueden realizar consultas con el objetivo de recuperar la información de la tabla de la que disponemos de manera muy sencilla.

En la página oficial de Query Language Reference, existe un apartado llamado “Setting the Query in the Data Source URL” en el cual se indica cual es la operación o consulta que se quiere realizar mediante lenguaje Query en la casilla de la izquierda, y este al clicar el botón “Encode”, nos lo transforma en un lenguaje para añadirlo en la URL de nuestra página de Google Drive compartida.

La URL del archivo almacenado en Google Drive es la siguiente:

<https://docs.google.com/spreadsheets/d/18T0kZWzCrcctjeVuXygUF1MHkzfUX4QU7TmID4qhNY/edit#gid=0>

Tenemos que modificarlo, quitándole la última parte, ya que solamente indica que la página se irá editando, y se le añadirá “gviz/tq?tqx=out:html&tq= “, indicándose con la adición de “html” el tipo de archivo que obtenemos y “=” sirve para indicar después la consulta que se quiere realizar. Una vez hecho esto, nos queda la siguiente expresión, que será nuestra URL:

<https://docs.google.com/spreadsheets/d/18T0kZWzCrcctjeVuXygUF1MHkzfUX4QU7TmID4qhNY/gviz/tq?tqx=out:html&tq=>

A continuación se realizan las tablas que se creen convenientes para facilitar la programación posterior en R. Para ello, hay que tener en cuenta que tenemos dos columnas en la tabla que se va a manipular, A (fecha) y B (dato de corriente).

En primer lugar, se deben pasar estos datos a valores de potencia consumida por el usuario. Los cálculos se realizarán suponiendo que la tensión es de 230V y el factor de potencia del consumidor es de 0.9. En los casos en los que se requiera de un valor de energía, el cálculo se realiza sumando las potencias que se obtienen en el intervalo de tiempo deseado y dividiéndolo entre 6, ya que se tienen 6 datos por cada hora.

$$\text{Potencia} = V \times I \times \text{Factor de potencia} = 230 \times I \times 0.9$$

$$\text{Energía} = \frac{V \times I \times \text{Factor de potencia}}{t} = \frac{1}{6} \times 230 \times I \times 0.9$$

1. Tabla 1: La energía horaria en Wh

Al crear esta tabla se obtienen 24 datos de energía por cada día, es decir, 8760 datos de energía por cada año. En este caso, los datos aparecen en 5 columnas, el año, el mes, el día, la hora y el dato de energía (Wh) respectivamente, de forma separada. Para la realización de esta primera tabla, el código utilizado es el siguiente:

```
select year(A), month(A), day(A), hour(A), 230*0.9*(sum(B))/6
group by year(A), month(A), day(A), hour(A)
```


Tal y como se puede ver en la siguiente figura, insertando este código en la casilla de la izquierda y pulsando el botón “Encode” se obtiene la instrucción que se le debe añadir a nuestra URL.

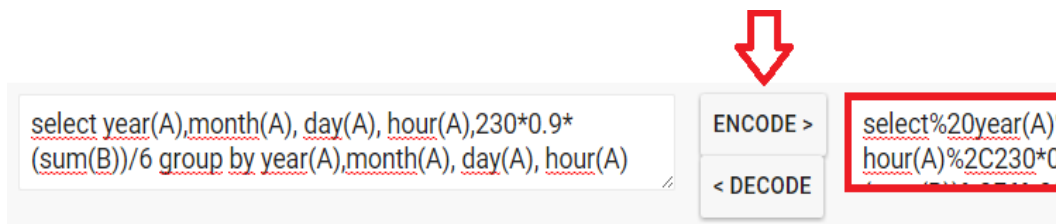


Figura 8: Ejemplo de cómo obtener el código adecuado para añadirlo a nuestra URL

Así, el código obtenido es el siguiente:

```
select%20year(A)%20month(A)%20day(A)%20hour(A)%20230*0.9*(sum(B))/6%20group%20by%20year(A)%20month(A)%20day(A)%20hour(A)
```

Añadimos este código a nuestra URL, obteniendo una nueva URL que se introduce en la barra superior del navegador:

```
https://docs.google.com/spreadsheets/d/18T0kZWzCrcctjeVuXygUF1MHkzfUX4QU7TmID4qhNY/gviz/tq?tx=out:html&tq=select%20year(A)%20month(A)%20day(A)%20hour(A)%20230*0.9*(sum(B))/6%20group%20by%20year(A)%20month(A)%20day(A)%20hour(A)
```

year()	month()	day()	hour()	quotient(product(product(230)0.9)sum)6()
2014	0	1	0	90.856865592
2014	0	1	1	113.68171952249999
2014	0	1	2	128.6063438985
2014	0	1	3	135.968781312
2014	0	1	4	118.452232863
2014	0	1	5	129.249874779
2014	0	1	6	127.09117279499999
2014	0	1	7	132.22537564799998
2014	0	1	8	131.040546771
2014	0	1	9	116.57798413799999
2014	0	1	10	106.5963898005
2014	0	1	11	115.475125191
2014	0	1	12	101.38445327250001
2014	0	1	13	99.13756259550001
2014	0	1	14	101.50571787450001
2014	0	1	15	115.32550083
2014	0	1	16	102.309703701

Figura 9: Tabla de energía horaria con Query Language reference

Mediante esta tabla es fácil acceder a los datos horarios de energía con su correspondiente fecha.

2. Tabla 2: La energía diaria en Wh

En esta tabla se tienen 365 datos de energía por cada año, que se van acumulando a lo largo de los años. En este caso, los datos aparecen en 4 columnas, el año, el mes, el día y el dato de energía (Wh). Para la realización de esta tabla, el lenguaje utilizado es el siguiente:

```
select year(A), month(A), day(A), 230*0.9*(sum(B))/6 group by year(A), month(A), day(A)
```

Actuando de la misma forma que en el caso anterior, es decir, transformándolo al igual que en la figura 10 obtenemos el código que hay que añadirle a la URL inicial:

```
select%20year(A)%20month(A)%20day(A)%20230*0.9*(sum(B))%2F6%20group%20by%20year(A)%20month(A)%20day(A)
```

Así la segunda URL generada es:

```
https://docs.google.com/spreadsheets/d/18T0kZWzCrncctjeVuXygUF1MHkzfUX4QU7TmlD4qhNY/gviz/tq?tqx=out:html&tq=select%20year(A)%20month(A)%20day(A)%2C230*0.9*(sum(B))%2F6%20group%20by%20year(A)%20month(A)%20day(A)
```



year()	month()	day()	quotient(product(product(230(0.9))sum)6()
2014	0	1	2765.8946786249985
2014	0	2	16117.206594112495
2014	0	3	13983.021765357
2014	0	4	13588.737437301003
2014	0	5	6947.846514991496
2014	0	6	2966.4000000465003
2014	0	7	27438.728609775004
2014	0	8	26341.669406991015
2014	0	9	24024.218030031
2014	0	10	14453.497516707006
2014	0	11	14459.011435667999
2014	0	12	16630.203484717502
2014	0	13	15665.8752506655
2014	0	14	24982.502733517485
2014	0	15	26526.0819701235
2014	0	16	23123.358827463002

Figura 10: Tabla de energía diaria con Query Language Reference

3. Tabla 3: La energía mensual en Wh

En esta tabla se recogen los datos mensuales de energía, es decir, se tienen 12 datos para cada año. En este caso se reduce el número de columnas a tres, ya que se expresa el año, el mes y la energía consumida (Wh). Para esto el código creado es el siguiente:

```
select year(A),month(A),230*0.9*(sum(B))/6 group by
year(A),month(A)
```

Siguiendo con el mismo procedimiento que en los casos anteriores, y pulsando el botón “Encode” después de insertar el código en casilla de la izquierda en la página de Query, se obtiene el texto que hay que añadirle a la URL:

```
select%20year(A)%2Cmonth(A)%2C230*0.9*(sum(B))%2F6%20group%20by%20
year(A)%2Cmonth(A)
```

Y así obtenemos la URL que debemos insertar en el navegador para poder ver la tabla, que se vería de una forma similar a los casos anteriores:

```
https://docs.google.com/spreadsheets/d/18T0kZWzCrncctjeVuXygUF1MHkzfUX4QU7T
mID4qhNY/gviz/tq?tx=out:html&tq=select%20year(A)%2Cmonth(A)%2C230*0.9*(sum
(B))%2F6%20group%20by%20year(A)%2Cmonth(A)
```

4. *Tabla 4: La energía anual en Wh*

En la siguiente tabla se visualizará el computo de energía consumida durante todo el año, teniendo dos únicas columnas, donde en la primera de ellas aparece el año y en la segunda la energía consumida (Wh). El código que se debe emplear es:

```
select year(A),230*0.9*(sum(B))/6 group by year(A)
```

Que una vez transformado queda de la siguiente forma:

```
select%20year(A)%2C230*0.9*(sum(B))%2F6%20group%20
```

Y por tanto, la cuarta URL es la siguiente:

```
https://docs.google.com/spreadsheets/d/18T0kZWzCrncctjeVuXygUF1MHkzfUX4QU7T
mID4qhNY/gviz/tq?tx=out:html&tq=select%20year(A)%2C230*0.9*(sum(B))%2F6%20
group%20by%20year(A)
```

5. *Tabla 5 : Último dato de potencia diezminutal*

Se trata de una tabla donde únicamente se recoge el último dato diezminutal con toda su información sobre el instante en el que se da esa información. Por tanto aunque únicamente tenga una fila, dispondrá de 7 columnas, en las que respectivamente nos darán la información sobre, el año, mes, día, hora, minutos, segundos y potencia instantánea en W. El código para ello es:

```
select
year(A),month(A),day(A),hour(A),minute(A),second(A),(B*230*0.9
) order by a desc limit 1
```

Que al transformarlo queda de la siguiente forma:

```
select%20year(A)%20month(A)%20day(A)%20hour(A)%20minute(A)%20second(A)%20(B*230*0.9)%20order%20by%20a%20desc%20limit%201
```

Por tanto el último URL es el siguiente:

```
https://docs.google.com/spreadsheets/d/18T0kZWzCrcctjeVuXygUF1MHkzfUX4QU7TmlD4qhNY/gviz/tq?tqx=out:html&tq=SELECT%20year(A)%20month(A)%20day(A)%20hour(A)%20minute(A)%20second(A)%20(B*230*0.9)%20ORDER%20BY%20A%20DESC%20LIMIT%201
```

Estas son las cinco tablas que se han decidido usar para la realización del programa, es decir, la lectura de datos y su representación de una forma útil para el usuario. Al ir cambiando los datos de google drive, es decir, añadiendo un dato nuevo cada diez minutos, estas tablas también van cambiando.

Una vez hecho esto, ya se tienen los enlaces de internet, totalmente públicos a los que se va a acceder desde el programa realizado por R.

6.2 DISEÑO DE LA PÁGINA WEB

6.2.1 Introducción: R, RStudio y Shiny

Para el diseño y desarrollo de la página web se hace uso de uno de los entornos más populares para la creación de aplicaciones del que dispone R, RStudio [15]. R es un proyecto de software libre de GNU, es decir, cumple con los siguientes requisitos. Por un lado, respeta la libertad de los usuarios y de la comunidad. Los usuarios de este tipo de softwares tienen la libertad de ejecutar el programa como lo deseen, la libertad de estudiar cómo funciona el programa y cambiarlo para que haga lo que él quiera, teniendo para ello acceso al código fuente. Además dispone de total libertad para redistribuir copias para ayudar a otros usuarios y distribuir copias de sus versiones modificadas, ofreciendo así la oportunidad a toda la comunidad de beneficiarse de esas modificaciones.

Todas estas condiciones son fundamentales para que un software sea libre, por tanto, queda claro que un software libre es una cuestión de libertad y no de precio.

R se puede definir desde dos puntos de vista, por un lado se trata de un lenguaje de programación y por otro lado un entorno de trabajo. Estos dos están orientados fundamentalmente al cálculo estadístico y a la generación de gráficos. En lo que se refiere al lenguaje de programación, proporciona una amplia variedad de recursos para la creación de gráficos, el trabajo con ellos y el análisis estadístico. En cambio, como entorno de trabajo, se entiende como un sistema planificado y coherente que proporciona una serie de herramientas para manipulación, cálculo y representación gráfica de datos.

RStudio es un entorno de desarrollo integrado (IDE) para R y este se puede ejecutar igual que R a través de distintas plataformas como Windows, Mac o Linux, y además desde la misma web usando RStudioServer.

Para empezar a trabajar en el diseño de la página es imprescindible descargar en primer lugar el programa R desde la página oficial de R [16]. Una vez hecho esto, se debe instalar el programa R-Studio desde su sitio oficial [17]. Así se tiene el entorno de trabajo que se desea, que cuando arranca tiene el siguiente aspecto.

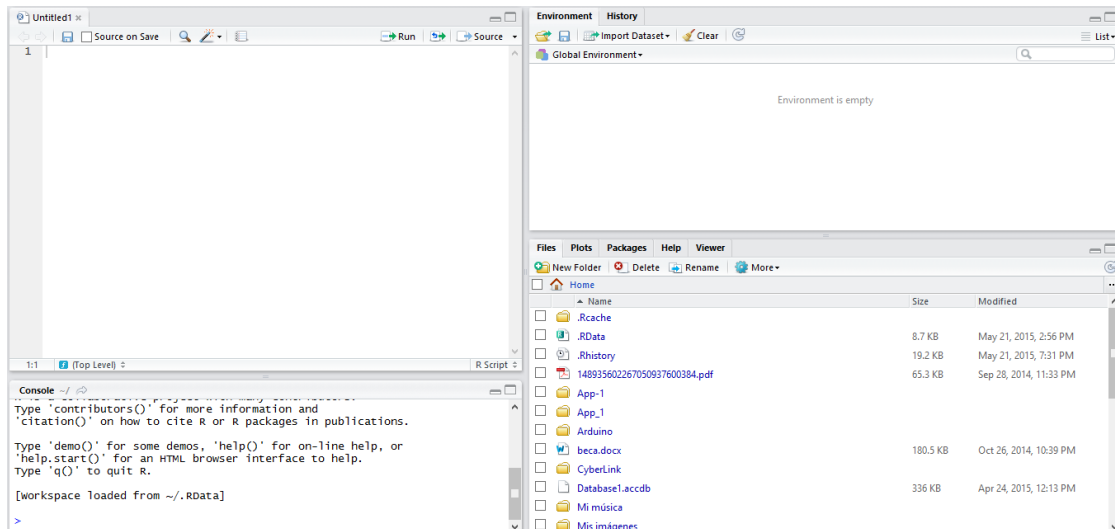


Figura 11: Aspecto de RStudio

RStudio es un entorno más cómodo y fácil de usar que R ya que es más amigable y visual. En la figura 11 se pueden ver las diferentes partes que encontramos en pantalla al abrir el programa. En la parte superior izquierda está el editor o el fichero que se encuentra abierto, donde se escriben las funciones que se quieren realizar para luego ejecutarlas. La parte superior derecha, Environment-History, es donde se muestran las funciones ejecutadas y las últimas acciones realizadas. Tal y como podemos ver, en la parte inferior izquierda aparece la consola, donde podemos realizar operaciones o visualizar los errores que tenemos en nuestro editor cuando ejecutamos o compilamos nuestro programa. Por último, en la parte inferior derecha, se encuentran las pestañas de Files-Plots-Packages-Help, que nos ayudan sucesivamente a encontrar archivos, representar gráficas, consultar paquetes y descargarlos. La pestaña help es para obtener ayuda específica de R, pero no es el único lugar donde podemos encontrar información, ya que también se puede acceder a información sobre RStudio desde el menú de ayuda.

Para poder representar todas las gráficas y cálculos que se quieren realizar con diferentes datos en una página web, es indispensable instalar la librería Shiny. Este es un paquete que hace más fácil construir páginas web interactivas. En la página web oficial de Shiny [18] hay un tutorial de 11 clases que resulta muy interesante y de gran ayuda.

Este paquete o librería no es el único que se debe instalar en RStudio, ya que a lo largo del trabajo, debido a las necesidades que han surgido, se ha necesitado de los siguientes paquetes: RCurl, XML, data.table, stringi, xtable, bitops, repmis, bitops, httr, gsheets.

El principio básico de RStudio para poder crear una aplicación Shiny (Shiny app) es la necesidad de generar dos editores o scripts. Uno de ellos será el “user interface”, que lo guardaremos como ui.R. En este archivo se determina el diseño y la apariencia de la página web, introduciendo diferentes pestañas, imágenes, colores y tipos de letras. El otro archivo necesario para el funcionamiento de la aplicación es el que se guarda como server.R y en este, se introducen las acciones necesarias para que el ordenador sea capaz de generar la web. En este archivo se realiza la lectura de los datos que se quieran analizar y se realizan todo tipo de cálculos. En este mismo script, se generan diferentes tipos de salidas como por ejemplo datos, gráficas y tablas para que luego se puedan mostrar en la app. Para que se visualicen las salidas en la web es necesario indicar en el script ui.R en qué lugar deben mostrarse los mismos. A continuación se muestra el aspecto que tendrán estos dos archivos:

```

1 library(shiny)
2
3 setwd("H:\\ejemplo_expli")
4
5 shinyUI(fluidPage(
6
7   titlePanel("TÍTULO"),
8
9   sidebarPanel(
10     wellPanel(
11       h4("Lo que escribamos en wellPanel aparecerá aquí,
12         y lo que escribamos en mainPanel, fuera del recuadro")),
13
14     mainPanel()
15 ))
16
  
```

Figura 12: Aspecto del script ui.R

```

1 library(shiny)
2
3 setwd("H:\\ejemplo_expli")
4
5 shinyServer(function(input, output) {
6
7
8 })
  
```

Figura 13: Aspecto del script server.R

Tal y como podemos ver, las funciones para definir estos dos archivos son `shinyServer` y `shinyUI`, para los que es necesario cargar la librería `shiny` previamente. Además, podemos apreciar que se ha indicado mediante `setwd` el directorio donde queremos que se guarde cada archivo. Para que puedan trabajar a la vez, es indispensable que los dos script estén en el mismo lugar. En la siguiente figura se puede ver cuál sería el aspecto de la página web después de haber escrito lo que se muestra en las figuras 12 y 13 y haberle dado al botón “run app”.

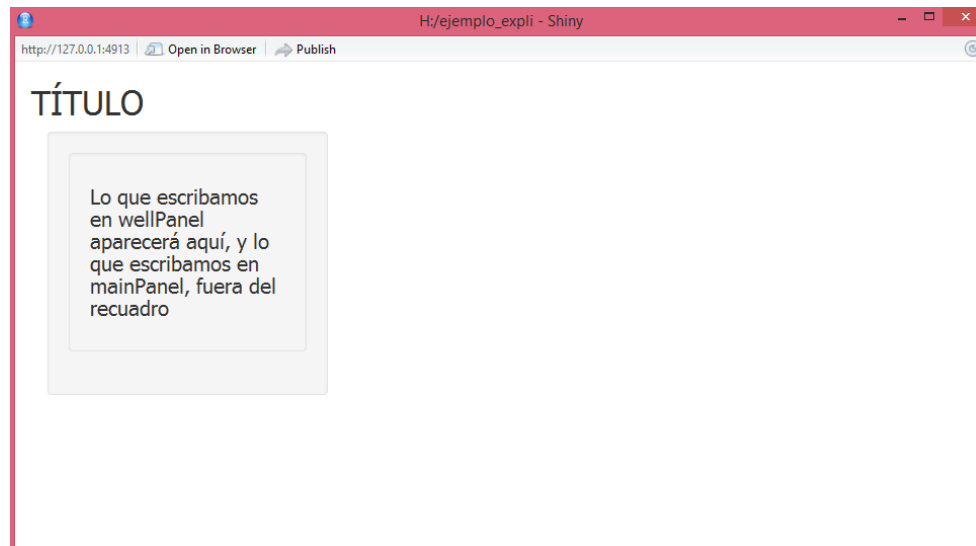


Figura 14: Aspecto de la página web

6.2.2 Acceso a los datos de las URL

Una de las cosas que han resultado de más dificultad, es la accesibilidad desde RStudio a los datos de las URLs. Para ello, tras una búsqueda intensiva en internet, y después de barajar diferentes posibilidades, la única opción que ha resultado válida es la propuesta en el blog “[revolutionanalytics](#)” [19], en el que nos facilita el código de dos funciones, “`readGoogleSheet`” y “`cleangoogleTable`”, con los que se podrá acceder a la tablas que se han creado en Google Drive y que después se han modificado.

De este acceso se encarga la primera de las funciones, la cual se encuentra adjuntada como *Anexo 1*. La segunda de ellas, *Anexo 2*, limpia y borra la información que no es relevante en la tabla, como por ejemplo los títulos de las columnas. No nos interesa que cuando lea la tabla nos borre la primera fila, ya que en ese caso, la línea que nos tomaría como título sería la fila 1, es decir, en el ejemplo que se muestra a continuación, los datos correspondientes al “01/01/2014”. Se representa mediante la figura 15 lo que ocurriría si utilizásemos la función `cleanGoogleTable` sin realizarle ninguna modificación.

year()	month()	day()	quotient(product(product(230()0.9())sum)6())
2014	0	1	2765.8946786249985
2014	0	2	16117.206594112495
2014	0	3	13983.021765357
2014	0	4	13588.737437301003
2014	0	5	6947.846514991496
2014	0	6	2966.4000000465003
2014	0	7	27438.728609775004
2014	0	8	26341.669406991015
2014	0	9	24024.218030031
2014	0	10	14453.497516707006
2014	0	11	14459.011435667999
2014	0	12	16630.203484717502

Annotations:

- Borraría esta línea* (Red arrow pointing to the header row)
- Esta fila la tomaría como título* (Black arrow pointing to the first data row)
- Esta fila sería la primera que se tendría en cuenta a la hora de realizar los cálculos en R* (Blue arrow pointing to the second data row)

Figura 15: Ejemplo de lo que ocurriría al aplicar la función cleanGoogleTable

Visto esto, la primera medida que se tomó fue eliminar dicha función, pero esto hizo que no funcionara la otra función, “readGoogleSheet”. Debido a esta observación se vio necesaria la interacción de las dos funciones para que diese los resultados esperados. Al observar detenidamente el código de la función problemática, se vio que en una parte del mismo, se hace referencia al título de la tabla, para que lo elimine. En este caso es interesante que no borre este título, para que así, la primera fila que tenga en cuenta sea la correspondiente a la real, y no como en el caso anterior, la segunda. Por ello se elimina esta parte del código, que corresponde al siguiente texto:

```

if(header && nrow(dat) > 1){
header <- as.character(dat[1, ])
names(dat) <- header
dat <- dat[-1, ]
}

```

Una vez hecho esto, las dos funciones necesarias para acceder a los datos de una URL están listas, y se deben guardar en el mismo directorio donde se encuentran los script ui.R y server.R. Para que estos dos archivos se ejecuten a la vez que server.R y ui.R, deben ser llamados desde el mismo. Para ello, el código utilizado es “source()”, y se escribe la ruta de acceso del archivo que se quiere cargar. Se debe tener en cuenta que según la versión de R que se tenga la separación para el nombre de la carpeta se debe hacer con una “/” o dos “//” barras.

Los datos que se tienen en las diferentes URLs, no son iguales a lo largo del tiempo, ya que tal y como hemos dicho, cambian cada 10 minutos. Por ello, es necesario tratarlos como datos tipo “reactive” para que estos vayan refrescándose cada in-

tervalo de tiempo que le indiquemos. Al ser datos diezminutales, sería suficiente con refrescarlo cada 10 minutos, es decir, 600000 ms, pero como no se va a conocer el instante en el que se suba el dato a la tabla de Google Drive, puede haber un error de hasta 9 minutos y 59 segundos, por lo que lo refrescaremos cada 5 minutos (300000ms), para que así el error disminuya. El código a emplear para que R sea capaz de coger las cinco tablas y limpiarlas para poder ser tratadas es el siguiente:

```
datos_entrada<-reactive({
  invalidateLater(300000,session)
  gdoc <- "URL DE LA HOJA DE GOOGLE DRIVE"
  elem <- readGoogleSheet(gdoc)
  m <- cleanGoogleTable(elem, table=1)
})
```

En la siguiente figura se puede ver cómo quedaría el código para el caso en el que la tabla llamada sea la de datos diarios.

```
#cargar datos de consumo de energía diaria(365 datos para cada año)
datos_diaros <-reactive({
  invalidateLater(300000,session)
  gdoc <- "http://docs.google.com/spreadsheets/d/18T0kZwzCrncctjeVu:
  elem <- readGoogleSheet(gdoc)
  m <- cleanGoogleTable(elem, table=1)
})
```



Figura 16: Ejemplo de código para cargar datos de energía diaria

Tal y como podemos observar, en lugar de poner “https:” se ha puesto “http:”, debido a que de la primera forma no se podía acceder al archivo. Al igual que en el caso de las barras “\”, esto se debe a la versión de Shiny. Al poner “session” dentro de invalidateLater, es indispensable indicar al principio de la función ShinyServer que además de las entradas (input) y salidas (output), tendremos que refrescar la sesión (session). Quedando de la siguiente forma:

```
shinyServer(function(input,output,session) {...})
```

Si quisiéramos ver la tabla que se ha generado en la figura 16 en la página web, en primer lugar tendríamos que definirla como una salida en el script server.R de la siguiente forma:

```
output$tabla_salida <-renderTable({
  datos_entrada()
})
```

Para luego poder llamarla desde ui. R:

```
tableOutput('tabla_salida')
```

6.2.3 Aspecto de la página web

Tal y como se ha especificado en el apartado del planteamiento de requisitos que debe cumplir la página web, debe ser de fácil uso para el usuario. Además, como es interesante que el cliente pueda acceder a la web desde cualquier dispositivo electrónico con conexión a internet, es importante que la información quepa en un espacio reducido. Esto se debe a que en el caso de un Smartphone, al tener una pantalla relativamente pequeña, el usuario se tendría que desplazar mucho en la página para conocer toda la información, además de dificultar y ralentizar el proceso de búsqueda del dato que le interesa en cada instante.

Por todo esto, surge la necesidad de crear diferentes pestañas, para tener una información ordenada y sencilla de encontrar. Se cree conveniente crear seis apartados diferentes, cuyos títulos serán los siguientes: consumo diario, consumo mensual, consumo anual, datos significativos, factura y consejos.

Además en las tres últimas pestañas, tendremos varios subapartados. En el caso de los datos significativos, habrá cuatro subpestañas: mes del año de mayor consumo, mes del año de menor consumo, día del año de mayor consumo y día del año de menor consumo. Por otro lado, en la factura también podremos distinguir dos partes, en una podremos ver la factura del mes actual y en la otra, a factura del mes anterior. Por último, en los consejos que se pretenden dar a los usuarios, habrá tres tipos de consejos que se explicarán en distintas pestañas: ahorro de luz, ahorro en la cocina y electrodomésticos. En la siguiente figura se puede ver un esquema para facilitar la comprensión de los apartados de la página web:

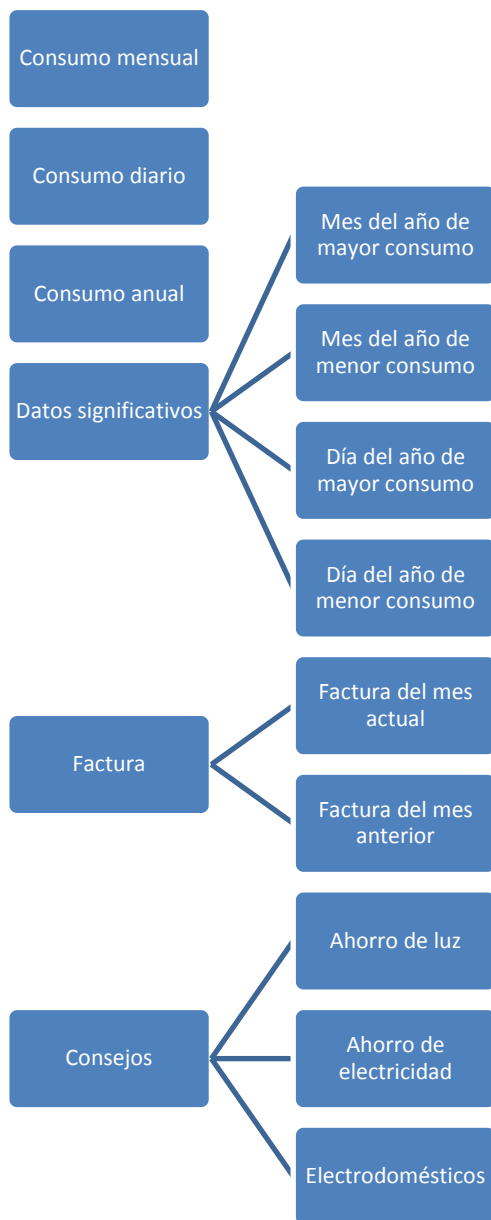


Figura 17: Esquema de los apartados de la página web.

En cada una de estas pestañas la pantalla se divide en dos. El extremo izquierdo de la pantalla se reservará para expresar los datos más relevantes sobre el consumo o un pequeño resumen de los datos recogidos (esto quiere decir que se escribirán en el apartado “sidebarPanel((wellPanel())”). Por otro lado, las gráficas y la factura se representarán en el resto de la pantalla, es decir, mediante “mainPanel()”. Esto ocurre en todas las pestañas excepto en la de consejos, donde no dividiremos la pestaña debido a que no se cree necesario.

Para generar estos apartados para la web, solamente hay que modificar el script ui.R, ya que este es el que se encarga de la apariencia de la misma. La función necesaria para ello es:

```

navbarPage ("",
  tabPanel ("1-Consumo diario"),
  tabPanel ("2-Consumo mensual"),
  id="pestanas")

```

Para insertar información como puede ser texto o imágenes en el script ui.R, existen diferentes funciones que se encuentran en el tutorial de *Shiny*[13], que son los mismos que se emplean en HTML5:

<i>Función Shiny</i>	<i>Función HTML5</i>	<i>Descripción</i>
P	<p>	Un párrafo
h1	<h1>	Título principal
h2	<h2>	Subtítulo
h3	<h3>	Título de tercer nivel
h4	<h4>	Título de cuarto nivel
h5	<h5>	Título de quinto nivel
h6	<h6>	Título de sexto nivel
A	<a>	Link de internet
Br	 	Línea en blanco
Div	<div>	Parte de texto con estilo uniforme
Span		Parte interna de un texto con estilo uniforme
Pre	<pre>	Tipo de letra más fina
Code	<code>	Tipo de letra ordenador
Img		Imagen
Strong		Letra negrita
Em		Letra cursive
HTML	<HTML>	Directamente pasa los caracteres al código HTML.

Tabla 2: Funciones shiny para ui.R

La última función expresada en la tabla es HTML y se utiliza para acceder a partes ya existentes de otras páginas web y que se quieran utilizar en la web que se pretende crear. Así se tiene acceso por ejemplo a imágenes que se encuentren en la red sin tener la necesidad de descargarlas, guardarlas en el mismo directorio y llamarlas desde la función ui.R.

En la web “Conoce tu consumo”, la función HTML será empleada para insertar fotografías en los diferentes apartados de la aplicación, y además se usará para crear el encabezado de la página web. El encabezado de la página, además del título de la misma, tiene incorporada una imagen y las iniciales de la web, CC.

En lugar de crearlo con una imagen y guardarlo en el mismo directorio que el resto de archivos necesarios, se ha optado por subir este archivo a un foro [20] que se

dedica exclusivamente a almacenar imágenes de una forma pública. Esto se ha hecho para reducir el número de archivos guardados en el directorio. Así se obtiene el siguiente link para acceder a la imagen: <http://oi57.tinypic.com/r224jr.jpg>

Para escribir el texto que es necesario en la web, se utilizan diferentes tipos y tamaños de los citados en la tabla anterior.

A continuación se muestra el código necesario del script ui.R para representar tanto la foto de portada (encabezado) y las pestañas:

```
library(shiny)
shinyUI(fluidPage(
  tags$div(
    HTML("
      <img src= 'http://oi57.tinypic.com/r224jr.jpg' />
    ")),
  # pestañas
  navbarPage("",
    tabPanel("1- Consumo diario",
      sidebarPanel(
        wellPanel(),
        mainPanel()
      ),
    ),
    tabPanel("2- Consumo mensual",
      sidebarPanel(
        wellPanel(),
        mainPanel()
      ),
    ),
    tabPanel("3-Consumo anual",
      sidebarPanel(
        wellPanel(),
        mainPanel()
      ),
    ),
    tabPanel("4- Datos significativos",
      navbarPage("",
        tabPanel("Mes del año de mayor consumo",
          sidebarPanel(
```

Figura 18: Código necesario en ui.R para crear las pestañas.

Con este código el aspecto que tendría nuestra página sería el siguiente:



Figura 19: Aspecto de la página web

6.2.4 Estilo de la fuente: Archivo CSS

El lenguaje CSS, “Cascading Style Sheets”, se emplea para la presentación de un documento estructurado escrito en HTML o XML [21]. La idea de este tipo de archivo es generar un código que separe la estructura del documento en cuanto a la presentación, es decir, formato y estilo.

Se ha decidido crear una hoja CSS en el mismo directorio que el resto de archivos necesarios, para así luego poder llamarlo desde el script ui.R (este es el encargado de la apariencia de la página). Para la llamada del archivo se utiliza la función *includeCSS()*, en la que entre paréntesis únicamente hay que poner el nombre del archivo, sin hacer referencia a la carpeta a la que pertenece.

Por tanto, se genera un archivo llamado *estilo.css* (anexo 3) que se encarga de estructurar la aplicación. Únicamente se determinan los colores, tipos y tamaños de diferentes letras utilizados a lo largo del script ui.R: h1, h2, h3, h5 y body. Se trata de un archivo muy sencillo ya que el objetivo de este Trabajo Fin de Grado no es adquirir conocimientos sobre el tema y además el script ui.R permite realizar distintas modificaciones en el formato sin la necesidad de este archivo.

Con un script de este formato, se consigue un control centralizado de la presentación de un sitio web completo, agilizando de forma considerable la actualización del mismo. Por ejemplo si se define en el archivo *estilo.css* que el tipo de letra h1 sea cursiva, azul y de un tamaño determinado, todo lo que aparezca haciendo referencia a ese tipo de letra tendrá las mismas características.

6.3 CONTENIDO DE LA WEB

A continuación se muestra cómo se ha realizado el tratamiento de datos para que en cada una de las pestaña de la página web se tenga la información que resulte de interés.

Todos los cálculos necesarios para conocer la información que se quiere mostrar se realizan en el script server.R, y luego desde ui.R se indica dónde se quiere colocar dicha información, es decir en qué pestaña y en qué sitio concreto queremos que aparezca.

Todas las funciones y salidas se deben definir como reactivas, ya que la tabla de Google Drive irá cambiando cada diez minutos, por tanto las tablas de Query también, y la página debe ser capaz de coger los datos de la tabla actualizada cada vez que se refresque (300000ms).

En primer lugar se definirán las funciones para que calculen el número de horas, días, meses y años que han pasado desde que se empezó la medición de datos. A continuación se muestra un ejemplo del código necesario para hacerlo:

```
#calculo de numero de horas, dias, meses y años
numero_horas<-reactive({
  filas<-nrow(datos_horarios())
})
numero_dias<-reactive({
  filas<-nrow(datos_diarios())
})
numero_meses<-reactive({
  filas<-nrow(datos_mensuales())
})
numero_años<-reactive({
  filas<-nrow(datos_anuales())
})
```

Número de filas que tiene la tabla a la que se accede

Tabla a la que hay que acceder para conocer el dato deseado

Figura 20: Código para calcular el número de horas, días, meses y años

Con la definición de las funciones que calculan las variables mencionadas resulta mucho más fácil acceder a los datos de cada tabla.

6.3.1 Consumo diario, mensual y anual

En las primeras pestañas, que corresponden a la información sobre el consumo diario, mensual y anual, se sigue una estructura similar. En estas, en la parte izquierda de la pantalla se facilitan los datos significativos y en la parte derecha se muestran varias gráficas de fácil comprensión para el usuario. En la mayoría de los casos, estas son comparativas.

	<i>Datos diarios</i>	<i>Datos mensuales</i>	<i>Datos anuales</i>
<i>Datos significativos</i> (parte izquierda)	<p>DÍA ACTUAL:</p> <ul style="list-style-type: none"> - Último dato de potencia diezminutal (W). - Información temporal (año, mes, día, hora, minutos). - Consumo diario de energía (kWh). <p>DÍA ANTERIOR:</p> <ul style="list-style-type: none"> - Fecha del día anterior. - Consumo de energía (kWh). 	<p>MES DEL AÑO ACTUAL:</p> <ul style="list-style-type: none"> - Año actual. - Consumo mensual de energía (kWh). <p>MES DEL AÑO PASADO:</p> <ul style="list-style-type: none"> - Año anterior. - Consumo mensual de energía (kWh). 	<p>AÑO ACTUAL:</p> <ul style="list-style-type: none"> - Año actual - Valor del consumo de energía a lo largo del año actual (kWh). <p>AÑO ANTERIOR:</p> <ul style="list-style-type: none"> - Año anterior. - Valor del consumo de energía del año anterior (kWh).
<i>Gráficas</i> (parte derecha)	<ul style="list-style-type: none"> - Gráfica comparativa del día anterior y el día actual (datos horarios). - Gráfica de las últimas 24h (datos horarios) 	<ul style="list-style-type: none"> - Gráfica comparativa del mes del año actual con el del año anterior (datos diarios) 	<ul style="list-style-type: none"> - Gráfica comparativa del año actual con el anterior (datos mensuales)

Tabla 3: Información de las pestañas de datos diarios, mensuales y anuales

- **DATOS SIGNIFICATIVOS**

Para mostrar estos datos, en primer lugar hay que realizar la búsqueda del dato en las tablas. Por ejemplo, para el último dato de potencia se accederá a la primera fila de la séptima columna de la quinta tabla, y para la información sobre el instante en el que se da ese valor de potencia a las 5 primeras columnas, ya que los segundos no nos interesan:

year()	month()	day()	hour()	minute()	second()	product(product(230())0.9())
2015	2	11	21	50	0	120.15000008099999



Información temporal



dato de potencia

Figura 21: Acceso a la tabla

Los datos que son necesarios para realizar cualquier otro cálculo, hay que escribirlos como se ha hecho en la figura 20, pero si lo que se quiere es mostrar esos datos en la página web, es necesario guardarlos como una salida de la siguiente manera: “output\$nombre_de_salida “. Es fácil acceder a las tablas y mostrar así las medidas realizadas. A continuación se muestra el ejemplo de cómo guardar la variable de la información temporal del último dato de potencia como una salida:

```
output$fecha_hora<-renderText({
  año<-datos_potencia_diezminuta1()[1,1]
  aux<-datos_potencia_diezminuta1()[1,2]
  mes<-as.numeric(as.character(aux))
  nombre_mes<-switch(mes+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio",
    "Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")
  día<-datos_potencia_diezminuta1()[1,3]
  hora<-datos_potencia_diezminuta1()[1,4]
  if (hora<10){
    hora1=paste("0",hora)
  }
  else(hora1=hora)
  minutos<-datos_potencia_diezminuta1()[1,5]
  if (minutos<10){
    minutos1=paste("0",minutos)
  }
  else(minutos1=minutos)
  fecha_hoy<-paste(día,"de",nombre_mes,"del",año, " ",hora1,":",minutos1)
})
```

Figura 22: Guardar la salida de la información temporal

Se puede ver, que lo único que cambia respecto de las variables de número de días, horas... que se han calculado en la figura 20 es la primera fila. Cabe destacar que no siempre será “renderText”, esto depende del tipo de salida, pudiéndose llamar así mismo, “renderTable”, “renderPlot” ...

Una vez hecho esto, estas salidas deben ser llamadas desde ui.R para ser mostradas en la web. Tal y como hemos dicho esta información se mostrará en la parte izquierda de la web, por lo que estos datos de interés para el clientes serán llamados desde sidebarPanel(mainpanel()) en el orden en el que se quieren mostrar.

Se generan de la misma forma todas las salidas de la parte correspondiente a los datos significativos, es decir, la parte izquierda, pero deben tenerse en cuenta varios aspectos en los casos de día, mes y año anterior. Resulta imprescindible limitar la

salida, ya que en los casos en los que no se disponga de ningún dato de esos días, debe aparecer esa información en pantalla.

```

output$dato_mes_año_pasado<-renderText({
  if (numero_meses()>12) {
    aux<-datos_mensuales()[numero_meses()-12,3]
    a<-as.numeric(as.character(aux))
    b<-round(a/1000,2)
    paste(b,"kwh")
  }
  if (numero_meses()<=12){
    aux<-paste("No hay datos del año anterior")
  }
})

```

Figura 23: Ejemplo sobre la limitación de la salida. Datos del mes del año anterior.

En la figura 23 se observa que la salida será “No hay datos del año anterior” en caso de que el número de datos mensuales que se tenga sea igual o inferior a los correspondientes a un año, es decir, doce. Se debe programar de forma parecida en todos los casos.

- **GRÁFICAS**

DATOS DIARIOS

En la parte derecha de la pantalla de esta pestaña se muestran dos gráficas (véase la figura 25). En la primera de ellas se compara el consumo del día anterior y del día actual con diagramas de barras, mostrándose la primera de ellas con barras rellenas con rayas negras y la segunda de ellas con un relleno azul. Para la realización de esta gráfica se debe acceder a la tabla de datos horarios, en la cual, a través de la cuarta columna se puede conocer el número de horas transcurridas el día actual y así coger solamente los datos de energía de esas horas (gráfica azul de la figura 25). Para los datos del día anterior se tienen que coger los 24 datos que se encuentran antes de los del último día.

Al igual que en el caso de los datos significativos, se debe tener en cuenta que si no hay datos del día anterior se deben mostrar solamente los datos del día actual, además de un aviso que lo indique.

A continuación se presenta el código necesario para la implementación de la gráfica y guardarla como salida:

```

#comparativa de los dos últimos días
output$grafica_comparativa1<-renderPlot({
  aux<-datos_horarios()[numero_horas(),4]
  hora<-as.numeric(as.character(aux))
  b<- datos_horarios()[(numero_horas()-hora):(numero_horas()),5]
  y<-as.numeric(as.character(b))/1000
  1 if (numero_dias()<3) {
    z<-datos_horarios()[(numero_horas()-hora):(numero_horas()),4]
    nombre_hora<-as.numeric(as.character(z))
    limite=max(y)+0.1
    barplot(y,col="blue",width=1,xlab="horas",ylab="Energía consumida (kwh)",border="blue",
            ylim=c(0,limite),xlim=c(0,28),names.arg=nombre_hora)
    title(main="consumo del día actual", col.main="blue", font.main=4)
  }
  2 if (numero_dias()>=3){
    c<-datos_horarios()[(numero_horas()-hora-24):(numero_horas()-hora-1),5]

    y_anterior<-as.numeric(as.character(c))/1000
    y_ant_max<-max(y_anterior)
    y_max<-max(y)

    if ((y_ant_max)>(y_max)){
      3 limite=y_ant_max+0.1
    }
    else limite=y_max+0.1
    barplot(y,col="blue",width=1,xlab="horas",ylab="Energía consumida (kwh)",border="blue",
            ylim=c(0,limite),xlim=c(0,28))
    par(new=TRUE)
    barplot(y_anterior,col="black",width=1,axes=FALSE,density=40,xlim=c(0,28),
            ylim=c(0,limite),names.arg=c("0h","2h","4h","6h","8h","10h",
            "14h","16h","18h","20h","22h"))
    title(main="Comparativa del consumo con el día anterior", col.main="blue", font.main=4)
    legend("topleft", c("Consumo hoy", "Consumo ayer"),fill=c("blue","black"))
  }
}

```

Figura 24: Gráfica comparativa del consumo de energía del día actual con el día anterior

En la figura 24 se pueden distinguir tres partes:

-**Parte 1:** Si el número de días es inferior a 3, es decir, solamente han pasado uno o dos días desde el comienzo de las medidas, únicamente se mostrará la gráfica de los datos del último día. Esto se debe a que un transcurso de menos de 48 horas dificultaría la comparación.

-**Parte 2:** Si el número de días transcurridos es igual o superior a tres, ya se tienen los datos de dos días completos, por lo que la comparación se puede realizar sin problemas.

-**Parte 3:** Se hace uso de este código en todas las gráficas comparativas, para así por ejemplo en este caso, poder limitar el rango de valores que toma el eje y, comparando los valores máximos de los dos días consecutivos.

La segunda gráfica de esta pestaña se encarga de dar información sobre el consumo en las últimas 24h, es decir, representa los últimos 24 datos de la tabla de datos horarios. Al igual que en el caso anterior, esta gráfica necesita diferenciar dos casos, y es que cuando se disponga de más de 24h no hay ningún problema en la representación, pero al comienzo de la toma de medidas, si el número de horas transcurridas es menor que 24, se deben tomar solamente esas horas transcurridas.

Tal y como hemos dicho anteriormente, las gráficas se colocan en la parte derecha, por lo que habrá que llamar a la salida desde el script ui.R , escribiéndolo en el maiPanel(). A continuación podemos ver el aspecto que tendría la primera pestaña de la web, llamada “Consumo diario”.

Por último, se programan dos códigos para que aparezcan dos avisos. Uno de ellos informa en el caso en el que no haya datos del día anterior y el otro avisa cuando el consumo de energía del día actual supera el 80% del consumo de energía del día anterior, siempre que se tengan datos de más de dos días.

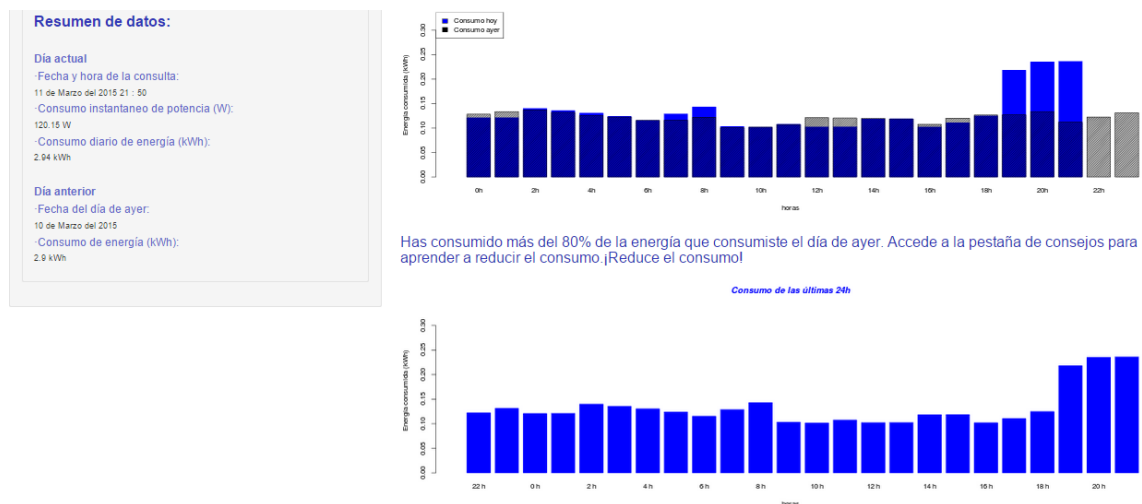


Figura 25: Aspecto de la pestaña de datos diarios

DATOS MENSUALES

En la parte derecha de la pantalla de esta pestaña se muestra una única gráfica. En la misma se compara el consumo del mes del año pasado con el de este año al igual que en el caso de los datos diarios.

Para la realización de esta gráfica se debe acceder a la tabla de datos diarios, en la cual, a través de la tercera columna se puede conocer el número de días transcurridos el mes actual y así poder coger solamente los datos de energía de esos días (gráfica azul de la figura 27). Para los datos del mes del año anterior, se deben coger los que se encuentran 365 o 366 posiciones más arriba en la tabla, dependiendo del número de días que tenga el año. Para saber el número de días que tenía el año pasado se crea una tabla de frecuencias, para saber cuántas veces aparece cada año en la tabla de datos diarios. La función empleada para ello es “table()”, y se debe tener en cuenta que mediante esta función se consigue una tabla (véase la tabla 4), pero que al guardar esos datos solo guarda la segunda columna, como si fuese un vector, no guarda la información de la primera columna.

2014	365
2015	71

Tabla 4: Tabla de frecuencias. Información sobre el número de días de cada año

En este caso también se deberán incluir condiciones para que cuando el número de meses transcurridos desde la iniciación de la toma de datos sea inferior a 12, solamente dibuje los meses del año actual.

Los avisos que se dan en esta pestaña son los mismo que en la pestaña de datos diarios.

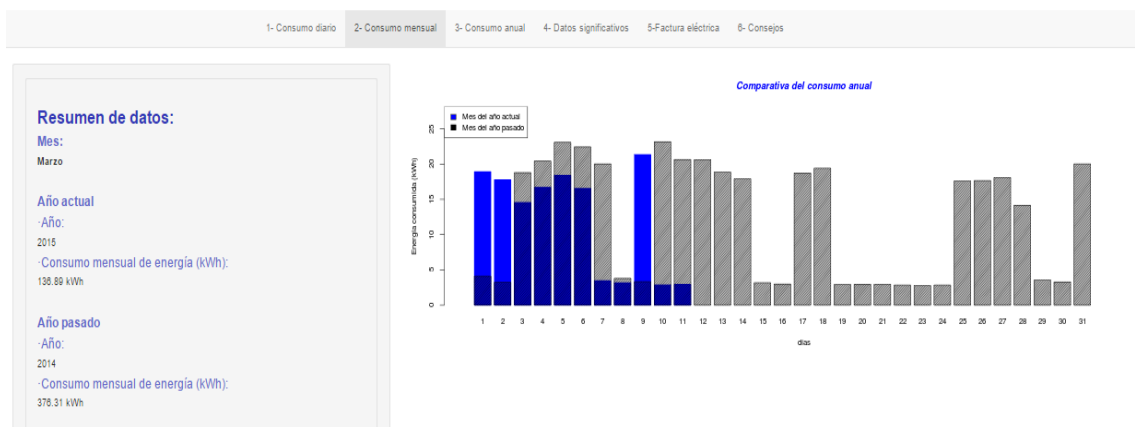


Figura 26: Aspecto de la pestaña del consumo mensual

DATOS ANUALES

La información necesaria para generar la gráfica de esta pestaña, se obtiene de las dos últimas filas de la tabla de la energía mensual en el que en la segunda columna tenemos los datos del mes de la consulta y en la tercera tenemos el dato sobre el consumo. Se programa de forma muy similar que en los casos anteriores, generando varios casos para que cuando no se tengan datos del año anterior solo represente los del último año.

No se ha considerado interesante poner un mensaje de aviso en esta pestaña para ver si se ha alcanzado el 80% del consumo del año anterior, ya que si en un mes en concreto el consumo del año actual es muy superior al del año anterior, ya nos varía mucho el dato anual y no es representativo para el momento específico de la consulta.

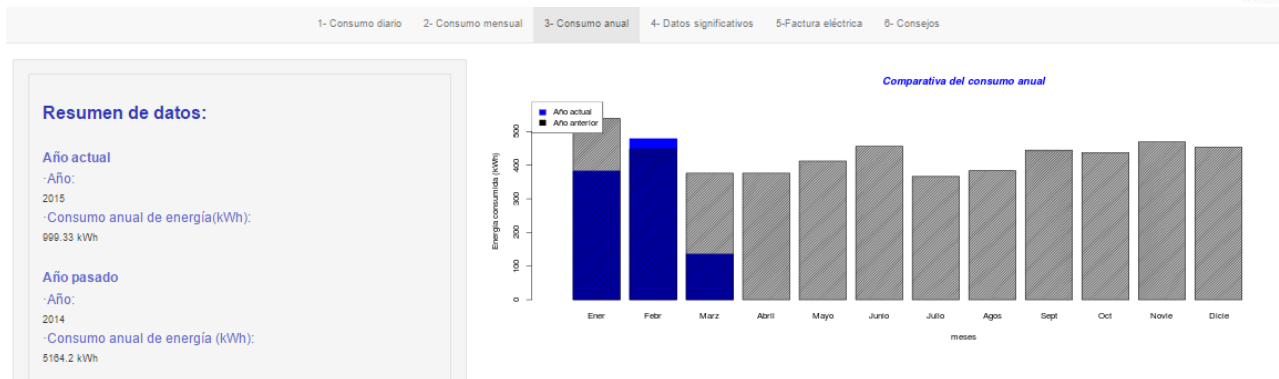


Figura 27: Aspecto de la pestaña consumo anual

6.3.2 Datos significativos

Esta pestaña es de gran importancia para el consumidor, ya que resulta de gran interés conocer la información sobre las fechas más características del año. Se ha considerado que estas fechas tan significativas son el mes de mayor consumo, el de menor consumo, el día de mayor consumo y el de menor consumo. Esto se debe a que la factura por ejemplo, se da para un intervalo de tiempo de un mes o dos, y por eso los dos primeros datos son de gran ayuda, para así poder ver cuando el consumo mensual es máximo o mínimo y por tanto la factura también.

Tal y como hemos especificado en el apartado sobre el aspecto que tendrá la web, en esta pestaña, habrá 4 subpestañas, cada una de ellas con el dato característico que se ha indicado. Todas estas disponen de dos partes, en la parte izquierda se indicará la información más relevante y en la parte derecha habrá un diagrama de barras.

- **MES DE MAYOR Y MENOR CONSUMO**

En las subpestañas del mes de máximo y mínimo consumo, en la parte izquierda de la pantalla, se muestra en primer lugar el mes en el que se da ese valor, la energía consumida en kWh y por último la factura eléctrica de ese mes, cuyo cálculo se explicará en el siguiente apartado. Para el mes de menor consumo, no se tiene en cuenta el mes en el que se están realizando las medidas, ya que esto haría que al no haber transcurrido todos los días del mes, la mayoría de las veces el valor mínimo fuese el del mes en el que nos encontramos en el momento de la realización de las medidas. Estos valores se dan con el acceso a la tabla de la energía mensual.

En la parte derecha de la pantalla se muestra la gráfica del consumo de energía de ese mes, con ayuda de la tabla de energía diaria y la tabla del número de días de cada año (véase tabla 4). Además se requiere de otra tabla en la que se recoja la frecuencia acumulativa de los días que han pasado hasta el inicio de cada año. Así, accediendo a la penúltima fila de la tabla se puede conocer el número de días que han

transcurrido desde que se comenzó la toma de datos hasta el 31 de diciembre del año anterior (véase la tabla 5).

Por último, es necesaria una última tabla en la que se guarden los datos de frecuencia acumulativa de los meses, es decir, nos aparece el número de días transcurridos hasta el inicio de cada uno de los últimos doce meses (véase la tabla 6). Para ello, en primer lugar se crea una tabla de frecuencias de los últimos doce meses, que los ordena desde enero hasta diciembre, sin tener en cuenta el último mes, y luego calcula la frecuencia acumulativa. Por esto, no son válidos los datos que se encuentran por debajo del dato del mes actual, ya que son valores del año anterior, pero esos datos no se utilizan, por lo que nos da igual. Así, al conocer el mes de mayor o menor consumo, se accede a la fila anterior a ese mes (se debe recordar que aunque la tabla es de dos columnas, a la primera de ellas no se puede acceder) y se puede saber cuántos días han pasado desde el comienzo del último año hasta el mes significativo.

$$\text{Día 1 del mes de máximo o mínimo consumo} = \text{Días hasta el 31 de diciembre del año anterior} + \text{Días transcurridos en el año actual hasta el día 1 del mes significativo}$$

La función necesaria para calcular una frecuencia acumulativa es “*cumsum(table())*”. Con el uso de esta función, se presentaba un error, ya que tal y como se ha comentado anteriormente, *table()* no se guarda como una tabla normal, solamente se guardan los datos de la segunda columna, por lo que al final es un vector de datos. Por esto, al calcular “*table()*”, se debe pasar al formato de tabla con “*data.table()*” y luego calcular la tabla de frecuencias acumulativas con “*cumsum()*”.

Las tablas de frecuencias acumulativas quedan de la siguiente forma:

2014	365
2015	435

Tabla 5: Tabla acumulativa días del año

1	31
2	59
3	90
4	120
5	151
6	181
7	212
8	243
9	273
10	304
11	334
12	365

Estos datos no nos sirven, solo los que son hasta el mes anterior a la consulta.

Tabla 6: Tabla acumulativa días de los meses

Siguiendo con el ejemplo que estamos teniendo en cuenta a lo largo de todo el proyecto, las tablas serían las representadas, y por ejemplo, para realizar la gráfica del mes máximo, como es el mes de febrero del 2015, accederíamos a la penúltima fila de la tabla 5 (año 2015-1) y a la primera fila de la tabla 6 (febrero-1). Sumando los valores representados en color azul, tendríamos la posición del día uno del mes de febrero.

Se sigue el mismo procedimiento para la representación de la gráfica del mes de menor consumo. En cualquiera de los casos, se debe limitar, que si el mes en el que se da ese máximo o mínimo valor es el mes de enero, considere que los días transcurridos el año en el que se realiza la consulta sea cero. Otra limitación que se debe hacer es que si la medición de datos no se realiza desde el uno de enero, tome solamente los meses en los que hay datos. Además para el caso en el que los datos no se tomen desde el inicio del mes, ese mes se descartará para el cálculo del mes de mínimo consumo, ya que sería muy probable que ese mes fuese el mínimo.

- **DÍA DE MAYOR Y MENOR CONSUMO**

En estas dos subpestañas se representan los días de máximo y mínimo consumo y al igual que en el resto de casos la información se da dividida en dos partes, en la parte izquierda de la pantalla se facilitan datos característicos como el valor del consumo en kWh y la fecha en el que se da ese valor. Esto se consigue accediendo a la tabla de consumo diario. Por otro lado, en la parte derecha de la pantalla se representa la gráfica con los datos horarios del día significativo.

Para la realización de la gráfica en este caso no se emplea el mismo procedimiento. En este caso, en primer lugar se calcula el número de horas que tiene la tabla de datos horarios, el número de horas que han transcurrido el último día y el número de días del año actual. Así una vez hecho es fácil acceder al primer dato de energía consumida el día de máximo o mínimo consumo:

$$00:00h \text{ del día de máximo o mínimo consumo} = \frac{\text{número de horas transcurridas desde el inicio de las medidas}}{\text{horas del último día}} - \left(\frac{\text{número de días del año actual}}{\text{la fila en la que se encuentra el dato significativo}} \right) * 24$$

En la siguiente imagen se puede ver el aspecto que adopta una de las pestañas citadas en este apartado, exactamente la del mes de máximo consumo.

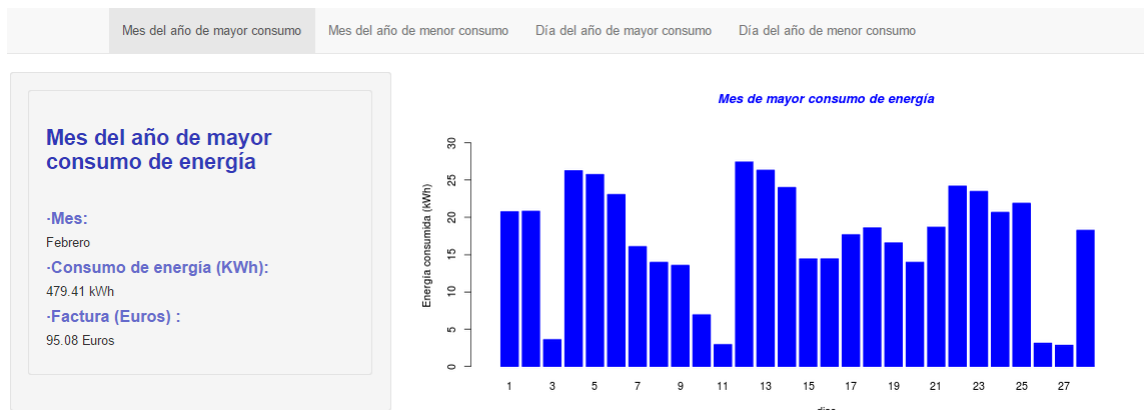


Figura 28: Aspecto de la pestaña del mes de máximo consumo

Al igual que en el resto de casos, para el cálculo del día máximo y mínimo consumo, también se generaran diferentes casos, para el caso en el que solo se tengan valores de un día, valores de un año, etc.

6.3.3 Factura eléctrica

- **CONCEPTOS IMPORTANTES SOBRE LA FACTURA ELÉCTRICA**

En una factura eléctrica existen conceptos de gran importancia que es indispensable conocer. En todas ellas se distinguen dos términos, el fijo, el cual es independiente del consumo que se realice cada mes y el variable, que depende del consumo de energía [22].

Dentro del *término fijo* se distinguen varios costes diferentes. Por un lado se encuentra el *término fijo de potencia*. Esta es la cantidad que se debe pagar mensualmente a la compañía eléctrica contratada aunque no haya consumido ningún kWh, esto se debe a que dicho pago da la garantía de poder conectar la potencia especificada en el contrato siempre que sea necesario. El importe a pagar se calcula multiplicando la potencia contratada por un coeficiente actualizado periódicamente por el gobierno y por el número de días que corresponden a la factura. Otro de los costes fijos es el *alquiler de equipos de medida* que se paga en el caso de que el consumidor no sea propietario del contador que tiene instalado en el hogar. En ese caso la distribuidora le cobra el alquiler con un precio mensual que viene detallado en la factura. Este coste es fijado por el Gobierno y el precio a pagar depende del tipo de contador. Existe otro término llamado *cuantía de peaje*, que es el coste que tiene la comercializadora por usar las redes de la distribuidora. Este término también viene fijado por el Gobierno y es obligatorio mostrar dicho precio en cada factura, pero suele incluirse dentro del coste fijo y variable, no se muestra de forma independiente.

En cuanto al *término variable*, se encuentra la *energía consumida*, la cual es la cantidad de kilovatioshora que consumimos en un periodo concreto. Esa cantidad de kWh consumidos en el periodo de tiempo indicado en la factura, se multiplican por el

precio que la comercializadora ha acordado con el cliente por kilovatiohora, y así se obtiene el importe total de ese término que se debe pagar. El *impuesto sobre la electricidad*, entra dentro de los impuestos especiales (alcohol, tabaco...) y se aplica al consumo y a la potencia. El comercializador se encarga de cobrar ese impuesto al cliente y de remitir esa cuantía al Gobierno. Esta cuantía está establecida por la legislación y se calcula de la siguiente forma:

$$\text{Impuesto sobre la electricidad} = 4.864\% \times (\text{coste consumo} + \text{potencia}) \times 1.05113$$

Además del impuesto de electricidad, a todo el coste a pagar se le aplica el 21% de IVA.

Hay un último coste que depende del contrato firmado por la comercializadora y el usuario, y es el coste de servicios de protección de pagos. Este es un término fijo, pero no es obligatorio.

Por tanto, en la factura aparecen los siguientes términos:

<i>Término</i>	<i>Cálculo</i>
Precio potencia contratada	Potencia contratada(kW) x días del mes x precio potencia €/kWdía
Consumo	Consumo de energía (kWh) x precio energía (€/kWh)
Impuesto electricidad	4.864% x (Precio potencia contratada + Consumo) x 1.05113
Servicio de protección de pagos	Servicio de protección de pagos (€/mes)
Alquiler de equipos	Alquiler de equipos (€/mes)
Total	Precio potencia contratada + consumo + impuesto electricidad + servicio de protección de pagos + alquiler de equipos
IVA (21%)	Total x 0.21
FACTURA TOTAL	TOTAL + IVA (21%)

Tabla 7: Cómo calcular los términos de la factura eléctrica

- **TIPOS DE TARIFAS**

Existen diferentes dos tipos de tarifas para contratar el suministro eléctrico desde el 1 de julio de 2009, las tarifas reguladas y las tarifas de mercado libre [23]. En el *mercado libre* el cliente elige la compañía y la tarifa que más le interesa dependiendo de sus necesidades. En este caso, el precio de la electricidad no está determinado por el estado, y por tanto, las compañías tienen total libertad para ofrecer los precios que consideren oportunos.

En cuanto a las *tarifas de último recurso (TUR)*, se trata de tarifas reguladas por el ministerio de Industria, Turismo y Comercio de forma trimestral, en los que estos se encargan de fijar el precio máximo que las comercializadoras pueden cobrar a

los clientes que contraten esta tarifa. El cliente puede contratar la TUR de cualquiera de las comercializadoras de último recurso. Cabe destacar que aunque muchas comercializadoras ofrecen esta tarifa o similares, las únicas que ofrecen este tipo de tarifas de forma oficial son las comercializadoras de último recurso (CUR).

En esta última tarifa, el Gobierno fija el término de potencia y el término de consumo de energía, que son las partes más significativas de la tarifa de la electricidad. El precio de la tarifa varía en función de la potencia contratada por el cliente y los periodos de facturación. Los consumidores que tienen derecho a este tipo de tarifa son los que tienen una potencia contratada igual o inferior a 10 KW.

Aunque todavía se use el término de tarifa de último recurso, desde el mes de enero de 2014 se denomina *Precio Voluntario al Pequeño consumidor (PVPC)*, y en la tabla 8 se muestran los precios más actualizados:

<i>Primer Trimestre 2015</i>	<i>Término Potencia €/kW y año</i>	<i>Término Energía €/kWh</i>		
<i>Tarifa</i>	<i>Periodo 1 €/kW y mes</i>	<i>Periodo 1</i>	<i>Periodo 2</i>	<i>Periodo 3</i>
<i>TUR sin discriminación horaria</i>	3.503619	0.124107	-	-
<i>TUR con discriminación horaria 2</i>	3.503619	0.148832	0.057995	-
<i>TUR con discriminación horaria 3</i>	3.503619	0.150812	0.71879	0.044146

Tabla 8: Precio de potencia y energía en función del tipo de tarifas [23].

La TUR sin discriminación horaria es aquella en la que se paga lo mismo por la energía consumida independientemente de la hora en la que se realice el consumo.

En la tarifa con discriminación horaria 2, se distinguen dos periodos con distinto precio del kWh a lo largo del día. El horario de discriminación horaria en invierno es de 12:00-22:00, periodo pico, donde la electricidad es más cara (periodo 1) y 22:00-12:00, el periodo valle, es decir, cuando la electricidad es más barata (periodo 2). En verano, cambian estos periodos siendo respectivamente de 13:00-23:00 y 23:00-13:00.

En el último tipo que se muestra en la tabla 8, se puede ver la opción de dividir las horas del día en tres periodos, los que se llaman periodo pico, periodo valle y periodo supervalle, en el cual la electricidad tiene el precio más bajo de todos. En este caso la distribución de las horas no depende de la época del año.

También se puede disponer de la posibilidad de contratar una tarifa con discriminación horaria en el mercado libre.

Además de las distintas posibilidades citadas, el 1 de octubre de 2015 arrancará una nueva tarifa horaria de la electricidad con la que se pague en función de lo que se consuma cada hora y de lo que cueste la electricidad en cada momento. Aunque esta tarifa se empiece a aplicar a partir de octubre, las comercializadoras tendrán disponible la información sobre el consumo por horas del cliente a partir de julio [24]. Esta tarifa será implantada a los consumidores que dispongan de un contador inteligente de electricidad, ya que este se encargará de realizar el cálculo de la factura en función del consumo y el precio marcado por el mercado mayorista en ese instante. El precio de la electricidad se puede consultar un día antes en la web de Red Eléctrica de España (REE) [25] a las 20.15h.

Este sistema de cobro se aplicará a todos los consumidores acogidos a la tarifa regulada o PVPC que dispongan de contadores inteligentes. Las comercializadoras de este tipo, tienen hasta octubre para hacer llegar a los clientes la primera factura “inteligente”.

Tras la implantación de la nueva tarifa horaria, cabe destacar que esta no sería la única opción tras la colocación de los nuevos contadores. Un consumidor con una tarifa de PVPC que no quiera tener la nueva tarifa horaria puede ir al mercado libre y contratar otro tipo de tarifa que se ajuste a sus requerimientos.

Aunque solamente se hayan explicado las tarifas para consumidores con una potencia contratada igual o inferior a 10kW, el resto de consumidores, los que tengan contratada una potencia superior, también tienen la posibilidad de contratar diferentes tarifas, pero en el mercado libre. No nos adentraremos en estas tarifas, ya que el proyecto trata de la medida del consumo de un hogar, es decir, una potencia contratada inferior a 10kW.

- ***FACTURA ELÉCTRICA DEL MES ACTUAL Y DEL MES ANTERIOR***

En la página web diseñada, resulta interesante mostrar dos facturas en paralelo mediante dos subpestañas diferentes, la del mes anterior y la del mes actual. Esto se debe a que muchas empresas comercializadoras entregan a los clientes la factura cada dos meses y con esta opción, puede ir consultando cuanto será su factura, tanto si tiene acordado con la comercializadora una entrega de la factura cada mes o cada dos meses. Además permite comparar el consumo realizado estos dos meses.

En este proyecto se considera que la empresa comercializadora es Endesa. y que el tipo de tarifa es el de mercado libre, sin discriminación horaria. Aunque resultaría interesante realizar el cálculo de la tarifa horaria implantada en los últimos días, por simplicidad, se escoge una tarifa del mercado libre sin discriminación hora. Los datos serán los siguientes:

Potencia contratada: 3.3 kW

Precio de la potencia: 0.115383 €/kW y día

Precio del consumo: 0.129862 €/kWh

Servicio de protección de pagos: 0.95 €/mes

Alquiler de equipos 0.58 €/mes

En la parte izquierda de la pantalla se presentaran los datos más interesantes sobre el contrato que se tenga firmado con la comercializadora, es decir, la potencia contratada y el precio de la potencia, además del año y el mes de la consulta y la energía consumida.

Por otro lado, en la parte derecha de la pantalla, se muestran los cálculos realizados para el cálculo de la factura mensual, y una gráfica con el consumo mensual de los últimos doce meses.

Esta última gráfica se ha considerado de interés debido a que las empresas comercializadoras suelen emplearla. Así, aunque no se tenga la factura eléctrica de los últimos doce meses, con este pequeño resumen del consumo de energía se puede visualizar la factura aproximada para cada mes.

A continuación se muestra el aspecto que adoptaría una de las subpestañas asociadas a la tarifa eléctrica.



Figura 29: Aspecto de la pestaña de la factura eléctrica

6.3.4 Consejos

Debido a la importancia de la reducción del consumo energético, a la hora de realizar la página web se ha creído conveniente crear una pestaña en la que se intente ayudar a reducir el consumo con su consiguiente disminución de la factura. Así, además de concienciar al consumidor en la importancia de un consumo eficiente, se contribuye en la reducción de consumo energético y por tanto en su impacto medioambiental.

El contador inteligente se encarga de informar al consumidor del instante en el que el consumo es mayor, por lo que se puede deducir cuál es el aparato que más consume.

Con la medición del consumo, se observa que aunque ningún aparato este aparentemente funcionando (por ejemplo por las noches), al mantenerlos en standby, es decir, en el modo de espera, consumen una gran potencia a lo largo del año. Al desconectar o apagar adecuadamente estos aparatos, el consumo se ve reducido, pero aunque todo este apagado el consumo no es cero. Esto se debe a que al mantener algunos aparatos apagados pero enchufados, tienen un consumo denominado “consumo fantasma” que se puede evitar de una sola forma, desenchufando el aparato. Uno de los consumos fantasma más significativo es el de la calefacción en verano o el aire acondicionado en invierno, así como las cocinas táctiles, ordenadores...

Para facilitar la búsqueda de los consejos que interesan en cada momento al usuario, se crean tres subpestañas en el apartado de consejos. En la primera de ellas se dan unas pautas para reducir el consumo de la luz, ya que la iluminación puede suponer el 9% del consumo eléctrico:

- Instalar interruptores divididos para poder encender únicamente las luces que sean necesarias.
- A la hora de cambiar las bombillas tradicionales, se aconseja hacerlo por bombillas de bajo consumo, que ahorran hasta el 80% de energía eléctrica y duran 8 veces más.
- Aprovechar la iluminación natural.
- Apagar las luces de las habitaciones que no estés utilizando.
- Con el mismo consumo las lámparas halógenas proporcionan más luz y su vida es de dos a tres veces mayor.
- En las estancias donde la luz debe estar encendida durante más tiempo, el uso de tubos fluorescentes permite un ahorro de energía de hasta el 80%, y duran de 8 a 10 veces más que las bombillas incandescentes. Las fluorescentes consumen más energía en el encendido, por lo que hay que tener en cuenta que no se estén encendiendo y apagando continuamente.

- Si dispones de iluminación exterior en tu jardín, conviene controlar su funcionamiento mediante un programador o interruptor crepuscular.
- Se deben utilizar colores claros en paredes y techos para aprovechar mejor la iluminación natural y reducir la artificial.
- La colocación de reguladores de intensidad luminosa de tipo electrónico ayudan al ahorro de energía.
- Se ahorra energía con detectores de presencia en garajes y zonas comunes para que las luces funcionen automáticamente.
- La iluminación localizada y la adaptación de la misma a las necesidades hace que se ahorre energía y se consigan ambientes más confortables.

La cocina es un lugar donde se derrocha mucha energía, por lo que un mayor cuidado en ella, hace que se consiga reducir el consumo energético y la factura. Por ejemplo, una nevera que no haya sido limpiada puede consumir el doble de electricidad. Por esto, es importante seguir las pautas de conservación de los electrodomésticos para reducir el consumo de los mismos.

Por otra parte, economizar el uso de la lavadora o del lavavajillas es de gran ayuda en la reducción del consumo eléctrico. Para ello, es imprescindible utilizarlos cuando estén completamente llenos. Además, hay que adaptar el ciclo adecuado para cada lavado, tal y como se indica en las instrucciones de estos electrodomésticos. Por todo lo mencionado, la segunda de las pestañas facilita la información para que siguiendo esos consejos se consiga un ahorro del consumo en la cocina considerable:

- Al cocinar, se aprovecha el calor residual si se apaga la cocina unos minutos antes de retirar el recipiente. (salvo para cocinas de inducción)
- Usar un recipiente con un diámetro ligeramente mayor que la zona de cocción y con fondo difusor de calor (salvo para cocinas de inducción), supone un ahorro en la energía consumida para cocinar del 20%.
- El uso de la olla a presión disminuye la energía consumida y se ahorra mucho tiempo.
- Conviene regular la temperatura del frigorífico y del congelador según las instrucciones del fabricante.
- El frigorífico y el congelador se deben instalar lo más lejos posible de los focos de calor.
- No se deben introducir alimentos calientes en el frigorífico, se deben dejar enfriar antes de introducirlos en el mismo.
- Una buena elección a la hora de comprar el frigorífico es determinante, conviene elegir modelos eficientes y ecológicos.
- Es importante eliminar la capa de hielo que se crea en el congelador, ya que así se consiguen ahorros del 30% de la energía.

- No se debe abrir el horno innecesariamente ya que se puede perder hasta un 20% del calor acumulado.
- Para poder controlar la pérdida de energía es importante usar un reloj programador en el horno.
- Es importante aprovechar al máximo la capacidad del horno, y esto se consigue cocinando el mayor número de alimentos a la vez.
- Si se va a cocinar durante más de una hora no es necesario precalentar el horno.
- Se debe aprovechar el calor residual del horno, por ello, se debe apagar el horno un poco antes de finalizar la cocción.
- Es interesante el uso de los hornos de convección, ya que favorecen la distribución uniforme de calor con su consiguiente ahorro de tiempo.
- El uso del microondas supone un ahorro energético de un 60-70% respecto de los hornos convencionales.
- Con el uso del lavavajillas se consume un 40% menos de energía que con agua caliente.
- Es importante el uso adecuado de un programa determinado para cada lavado con el lavavajillas y llenar el mismo en cada ocasión.
- Se deben colocar los utensilios de una forma adecuada para que el agua circule con facilidad.
- Usa los niveles de sal y abrillantador recomendados por el fabricante de lavavajillas.
- Compra un lavavajillas eficiente y ecológico.

Además de los electrodomésticos que se encuentran en la cocina, es importante un uso responsable y concienciado del resto de aparatos. Por esto, en el último apartado de la pestaña de consejos se dan las pautas necesarias para que se reduzca el consumo de los electrodomésticos:

- El uso de una temperatura moderada en la lavadora reduce mucho el gasto energético.
- Aprovechar al máximo la carga de la lavadora o secadora reduce el consumo energético y alarga la vida útil del electrodoméstico.
- Al igual que con el frigorífico y el lavavajillas, a la hora de comprar una nueva lavadora se deben escoger modelos eficientes y ecológicos.
- Un centrifugado de la lavadora a 1200 revoluciones, en vez de a 700 reduce el consumo de la secadora hasta un 20%.
- Elije ordenadores, impresoras u otros electrodomésticos con ahorro de energía.

- Ayudan en el ahorro de energía las lavadoras: termoeeficientes, con programas de media carga y con sonda de agua.
- Conviene aprovechar el calor del sol para secar la ropa.
- El uso de descalcificantes y una limpieza regular del filtro de la lavadora ayuda en ahorro de energía.
- Si tienes contratada la Tarifa con discriminación horaria, aprovecha las horas de descuento para poner la lavadora.
- Plancha grandes cantidades de ropa de una vez para aprovechar el calentamiento de la plancha.
- Optimiza el uso de los aparatos eléctricos.
- Apaga del todo (desenchufa) totalmente los televisores y equipos en ausencia superiores a 30 minutos.
- Apaga la pantalla del ordenador cuando no vayas a utilizarlo durante periodos cortos.
- El salvapantallas que menos consume es el de color negro.
- Los ladrones de conexión múltiple ayudan en el ahorro de energía.
- Un buen mantenimiento y la limpieza de los electrodomésticos prolonga su vida útil y ahorra energía.

Todos estos consejos son los facilitados por Iberdrola [26] en su web, que junto a otras recomendaciones sobre cómo reducir el consumo de agua caliente y calefacción, ayudan a un consumo más concienciado.

En las tres pestañas que se han creado, al final de los consejos, se representa un dibujo asociado a ellos. Estas son las pestañas más sencillas, ya que ni siquiera se necesita de dos partes diferenciadas en la pantalla, representándose todo en el mainPanel(). Todo el texto y los dibujos, se expresan en el script ui.R, ya que no necesitan de ninguna función para que aparezcan, simplemente se deben escribir los consejos. Las tres imágenes que aparecen en las pestañas, son de diferentes páginas de internet, por lo que, únicamente escribiendo la dirección a la que corresponden, aparece el dibujo. Por ejemplo, para la primera pestaña, el código a escribir para que aparezca el dibujo es el siguiente:

```
HTML ("
```

CONSEJOS

Disminución del consumo de la luz

- 1.- Colocar interruptores divididos para poder encender únicamente las luces que sean necesarias.
- 2.- Cambiar las bombillas tradicionales por bombillas de bajo consumo, que ahorran hasta el 80% de energía eléctrica y duran 8 veces más.
- 3.- Aprovechar la iluminación natural.
- 4.- Apagar las luces de las habitaciones que no estás utilizando.
- 5.- Las lámparas halógenas proporcionan más luz y su vida es de dos a tres veces mayor.
- 6.- En las estancias donde la luz debe estar encendida durante más tiempo, el uso de tubos fluorescentes permite un ahorro de energía de hasta el 80%, y duran de 8 a 10 veces más que las bombillas incandescentes. Las fluorescentes consumen más energía en el encendido, por lo que hay que tener en cuenta que no se estén encendiendo y apagando continuamente.
- 7.- Si dispones de iluminación exterior en tu jardín, conviene controlar su funcionamiento mediante un programador o interruptor crepuscular.
- 8.- Se deben utilizar colores claros en paredes y techos para aprovechar mejor la iluminación natural y reducir la artificial.
- 9.- La colocación de reguladores de intensidad luminosa de tipo electrónico ayudan al ahorro de energía.
- 10.- Se ahorra energía con detectores de presencia en garajes y zonas comunes para que las luces funcionen automáticamente.
- 11.- La iluminación localizada y la adaptación de la misma a las necesidades hace que se ahorre energía y se consigan ambientes más confortables.



Figura 30: Aspecto de la pestaña de consejos.

7. LA APLICACIÓN EN LA RED

Una vez realizada la programación de la aplicación, y comprobado su correcto funcionamiento mediante la simulación en RStudio a través de la herramienta Run app, se procede a la subida de la aplicación a la red, para así obtener la dirección de la página web deseada.

Antes de nada, se debe crear una cuenta en *www.shinyapps.io* para luego poder subir la aplicación creada. El nombre de usuario escogido para la cuenta, realiza la función de ser la primera parte del nombre de la web. En este caso el link asociado a la cuenta “consumoenergia” es el siguiente:

https://consumoenergia.shinyapps.io

Tras crear la cuenta de Shiny y conectarse a la misma, es suficiente con clicar en el botón “Publish” que se encuentra en la parte superior de RStudio.

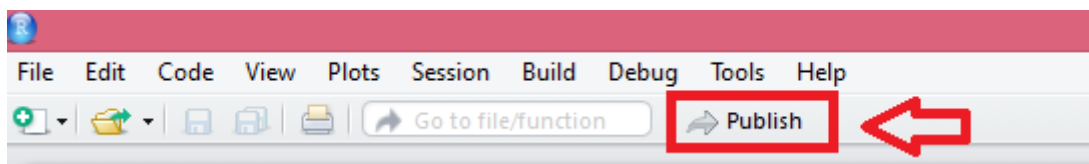


Figura 31: Subida de la aplicación a la red

Tras realizar esa acción, aparece en pantalla la ventana que se puede observar en la figura 32, en la que se muestran todos los archivos que se encuentran en el directorio. El nombre que se introduce en el apartado “Application” será el título de la página web, es decir, “conocetuconsumo”. Una vez clicado el botón de “publish”, comienza la subida de los archivos a la red.

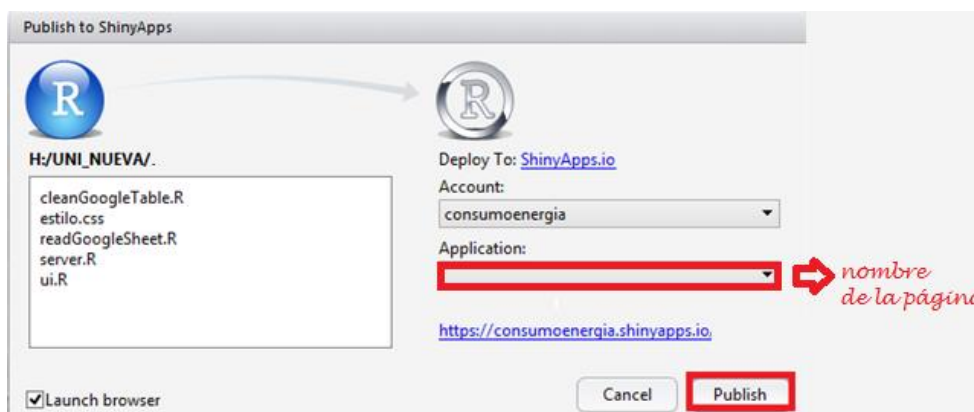


Figura 32: Ventana de los script necesarios para la implementación de la página web

Tal y como se ha explicado en el apartado 6.2.1, para que los scripts fuesen guardándose automáticamente se indicaba la carpeta mediante la función “*setwd()*”. En el caso de las funciones “*readGoogleSheet()*” y “*cleanGoogleTable()*” se había pro-

cedido de forma similar, guardando estas dos funciones en diferentes scripts pero en el mismo directorio que las funciones principales server.R y ui.R. Así, para que estas fuesen cargadas en la ejecución del programa debían ser llamadas desde server.R a través de la función “source()”. Por este motivo, al subir el programa a la red, este presenta un error, siendo incapaz de representar la web diseñada. Esto sucede porque una vez subidos los archivos a la red, estos no encuentran las carpetas que se citan en las funciones `setwd()` y `source()`.

La solución a estos problemas es sencilla, ya que únicamente hay que borrar esas líneas de código. Eso significa que el número de archivos del directorio se debe reducir a tres, introduciendo las líneas de código de los scripts “readGoogleSheet” y “cleanGoogleTable” al comienzo de server.R, ya que al eliminar las líneas de “source” no se está realizando la llamada a esas funciones.

Una vez reducido el número de archivos, se vuelve a publicar la aplicación pero se da el siguiente error:

```
The following lines contain absolute paths:
15: htmlTable <- gsub("^.*?(<table.*</table).*$", "\\1>", doc)
```

Tras la búsqueda de la solución a este error en la web [27], se realiza el cambio necesario para que la aplicación finalmente pueda ser subida a la red con éxito:

```
+doc <- content(GET(url), as="text")
-doc <- paste(readLines(url), collapse=" ")
```

Al sustituir el código +doc por -doc y volver a publicarlo, el código se ejecuta con éxito y la página <https://consumoenergia.shinyapps.io/conocetuconsumo> se visualiza en la red.



Figura 33: Aspecto de la página web en la red

8. VISUALIZACIÓN DEL CONSUMO ELÉCTRICO DE UN HOGAR

A lo largo de la realización de este proyecto se ha citado que los datos que se encuentran en la hoja de cálculo de Google Drive son variables, y que se añade un dato cada diez minutos, por lo que se deben tratar los datos en R considerándolos de tipo “reactive”.

El ejemplo mostrado hasta el momento mediante figuras no es el de una tabla con un número de filas variable. Esto se debe a que a la hora de realizar la programación de la web todavía las otras partes del proyecto, el sensado y el posterior cálculo de la corriente, no estaban implementadas, y por tanto no se disponía de datos reales. A pesar de ello, toda la programación se ha llevado a cabo teniendo en cuenta ese detalle, ya que resulta ser un factor determinante en el correcto funcionamiento de la web.

Una vez terminadas las cuatro partes constituyentes del proyecto inicial, se inicia la toma de medidas de la corriente mediante el Arduino y la RaspberryPi en la microrred de la Universidad Pública de Navarra, en la que se simula el consumo eléctrico de un hogar.

Para acceder a los datos de esas medidas, hay que realizar un único cambio en el código, ya que se debe introducir la página de Google Drive utilizada para cada caso, generando así los nuevos enlaces de Query Language Reference e introduciendo esos links en la parte del código correspondiente (analizado en el apartado 6.2.2).

Cabe destacar que a la hora de publicar la aplicación creada se han detectado varios errores, como por ejemplo, la necesidad de eliminar las gráficas comparativas en el caso de que no existan datos del día, mes y año anterior. Aunque estos detalles se hayan solucionado al final de la realización del trabajo, me han llevado bastante tiempo.

A continuación se muestra la página web en la que se visualizan los datos de la microrred con las medidas correspondientes al sensado y su posterior cálculo con el Arduino. El enlace en el que podemos encontrar estos datos actualizándose constantemente es:

<https://consumoenergia.shinyapps.io/microrred1>

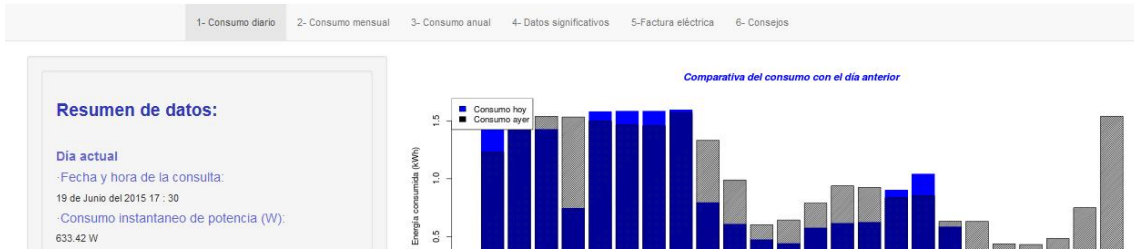


Figura 34: Aspecto de la pestaña consumo diario de las medidas en la microrred con el Arduino



Figura 35: Aspecto de la pestaña consumo mensual de las medidas en la microrred con el Arduino



Figura 36: Aspecto de la pestaña consumo anual de las medidas en la microrred con el Arduino

La mayor dificultad se ha dado a la hora de mostrar los nombres de los meses en el caso de que la toma de datos no se realice desde el mes de enero. Al final con un largo código, creando diferentes opciones, se ha conseguido que el nombre de los meses se represente de forma correcta en las gráficas de las pestañas de datos anuales y factura del mes actual.

De la misma forma se pueden visualizar los datos recogidos por la RaspberryPi en el siguiente enlace:

<https://consumoenergia.shinyapps.io/microrred2>

Cabe destacar, que si el Arduino o RaspberryPi falla, y en lugar de tomar un dato, deja la casilla de corriente vacía, aparece un error en la visualización de la página web, ya que la casilla no toma ningún valor (NA). Si más tarde comienza a tomar datos correctamente, para la realización de cálculos toma esas casillas blancas como cero, pero no se consigue representar la gráfica que contenga esos datos. A continuación se presenta un ejemplo de la página web cuando se tiene ese problema.

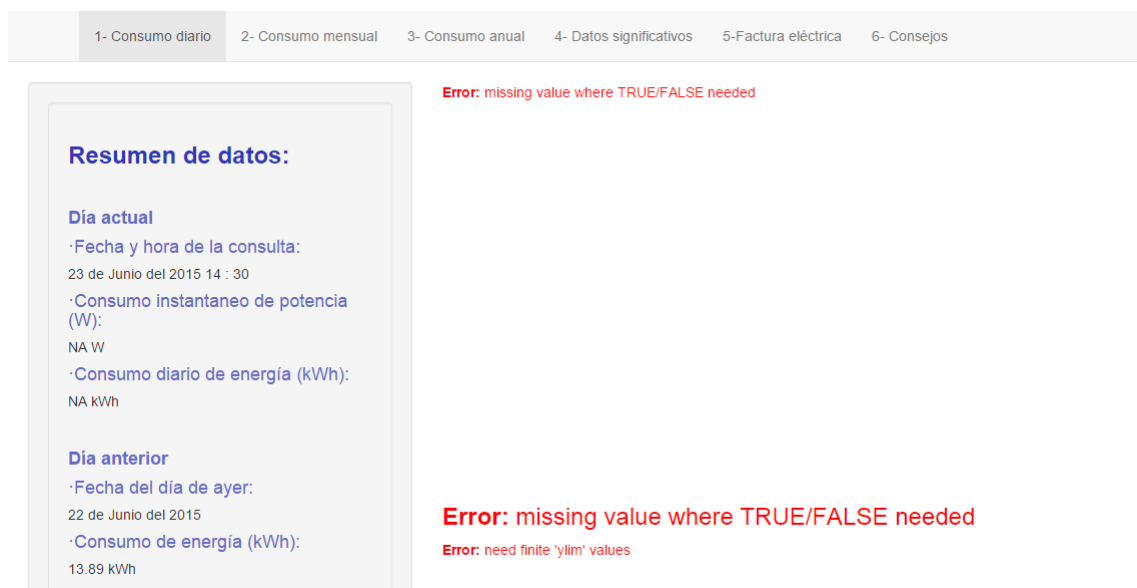


Figura 37: Error en la página web al fallar la Raspberry o el Arduino

9. CONCLUSIONES

El objetivo del proyecto general planteado fue diseñar y construir un contador para medir el consumo eléctrico de un hogar de dos formas distintas, mediante un Arduino y una RaspberryPi y la posterior visualización de los datos vía página web.

Como resultado al mismo, se puede concluir que los objetivos marcados han sido cumplidos con éxito en todos los casos aunque se hayan presentado diferentes problemas. En lo referente a los problemas que afectan directamente en la realización de este Trabajo Fin de Grado, en el caso de la programación del Arduino ha sido la subida de datos de corriente a la nube. Por lo contrario, en el caso de la RaspberryPi, el problema surgió al final, cuando Google deshabilitó la forma para autenticar mediante el correo electrónico y la contraseña, pero al final se encontró la alternativa adecuada.

La diferencia entre los precios de los dos contadores no es muy notoria, rondando los dos un precio de unos 70€, algo inferior en el caso del Arduino. La parte del sensado de la corriente tiene un coste para el Arduino de unos 40€ y para la RaspberryPi de 18€. El modelo más parecido al creado mediante este proyecto al que nos hemos referido a la hora de realizar el análisis del estado del arte, es el mostrado en la figura 3, "*Current Cost Envir*". Este tiene un coste de 90'75€, por lo que nuestro producto sería competitivo en el mercado, teniendo más posibilidades en lo que se refiere al precio, la RaspberryPi.

En cuanto a los objetivos específicos de este Trabajo Final de Grado: la visualización mediante una página web de los datos más característicos del consumo y de la factura eléctrica para un usuario no especializado en el tema, se cree que se ha cumplido con éxito. Los problemas más significativos que se han encontrado, como por ejemplo, el acceso a los datos almacenados en Google Drive y la posterior necesidad de modificar esas funciones halladas, se han solucionado en la mayoría de los casos con ayuda de la página web <http://stackoverflow.com/questions/>.

Otro de los problemas que no se ha citado hasta ahora, pero que resulta interesante, es que no es posible crear una salida en server.R y llamarla dos veces desde ui.R, siendo necesario crear dos salidas para expresar lo mismo en dos sitios diferentes de la página web.

Entre los puntos más destacados que se especificaron entre los objetivos se puede concluir que:

- Las gráficas y datos que se representan en las diferentes pestañas son muy características y de fácil comprensión.

- Los avisos sobre el consumo y la pestaña de consejos le proporcionan un aspecto de conciencia con el medioambiente.
- La pestaña de la factura eléctrica instantánea junto a la de consejos ayuda a la reducción del consumo para beneficio del usuario
- El reducido número de datos en cada pestaña facilita la visualización en “Smartphone” y “Tablet”.
- Rapidez de acceso a los datos: Al entrar a las páginas web creadas, el tiempo que le cuesta a los datos cargarse es de alrededor de 5 segundos, asique puede decirse que es bastante rápido.

Además de los objetivos citados, se ha adquirido un amplio conocimiento sobre la programación en R, Query y la librería Shiny.

Uno de los mayores inconvenientes que presenta la página, es que además de que la página se refresque, es decir, carga los nuevos datos cada cinco minutos adquiriendo la web un color más oscuro, cada cierto tiempo la página también adquiere un color más claro sin seguir ningún patrón de tiempo. Aun así los datos siguen viéndose sin mayor problema.

10. LÍNEAS FUTURAS

Observando los inconvenientes de la web creada, junto a nuevas posibilidades que se han creído interesantes a lo largo de la realización del proyecto, se ha creído oportuno citar las siguientes pautas a seguir en investigaciones futuras:

- Mejora del refresco de la página, reduciendo solamente el parpadeo al caso del refresco cada 5 minutos.
- Generar un código más elaborado pero reducido para los nombres de los meses cuando los datos no se miden a partir del mes de enero.
- Generación de más tablas en Query con un número menor de datos, para que R necesite menos tiempo para cargarlos y ocupen menos espacio. Esto facilitaría muchísimo la programación en R, reduciendo el número de líneas de código.
- Posibilidad de descargar la factura eléctrica desde la web.
- Posibilidad de introducir el intervalo de fechas sobre el que se quiere recibir información.
- Adecuar el cálculo de la factura de la electricidad a la facturación por horas que se establecerá a partir del mes de Octubre, ya que los hogares que dispongan de este tipo de contadores y tengan contratada una tarifa PVPC estarán obligados a este tipo de facturación.
- En lugar de crear una página web en el que pueda entrar cualquier persona, limitar su acceso con nombre y usuario.
- Buscar una solución para el caso en el que la RaspberryPi y el Arduino presenten algún fallo y dejen de subir muestras a la nube de almacenamiento

11. BIBLIOGRAFÍA

[1] Joseba Revuelta, “Monitorización del consumo eléctrico de un hogar: sensado y acondicionamiento de la señal”, Trabajo Final de Grado, Universidad Pública de Navarra, 2015.

[2] Víctor Erice, “Monitorización del consumo eléctrico de un hogar: Procesado de datos mediante Arduino”, Trabajo Final de Grado, Universidad Pública de Navarra, 2015.

[3] Aritz Legarrea, “Monitorización del consumo eléctrico de un hogar: Procesado de datos mediante RaspberryPi”, Trabajo Final de Grado, Universidad Pública de Navarra, 2015.

[4] The OWL: <http://www.theowl.com/>

[5] Efergy: <https://www.efergy.com/>

[6] Current Cost: <http://www.currentcost.com/>

[7] Engage Efergy: <https://engage.efergy.com/dashboard>

[8] Dropbox: <https://www.dropbox.com/>

[9] One Drive: <https://onedrive.live.com/about/es-es>

[10] Box: https://www.box.com/es_ES/home/

[11] Mega: <https://mega.co.nz/#info>

[12] Google Drive: https://www.google.com/intl/es_es/drive/

[13] Arturogoga:

<https://www.arturogoga.com/2015/02/19/comparacion-servicios-de-alojamiento-dropbox-vs-onedrive-vs-google-drive-vs-mega-vs-copy-vs-box/>

[14] Query Language Reference:

https://developers.google.com/chart/interactive/docs/querylanguage#Setting_the_Query_in_the_Data_Source_URL

[15] Blog Urcera: <http://blog.urcera.com/wordpress/?p=242>

Sistema operativo GNU: <http://www.gnu.org/philosophy/free-sw.es.html>

[16] The Comprehensive R Archive Network: <http://cran.at.r-project.org/>

[17] RStudio: <http://www.rstudio.com/>

[18] Shiny by RStudio: <http://shiny.rstudio.com/tutorial/>

[19] Revolutions:

<http://blog.revolutionanalytics.com/2014/06/reading-data-from-the-new-version-of-google-spreadsheets.html>

[20] Tinypic: <http://es.tinypic.com/>

[21] Norfipic: <http://norfipic.com/web/manual-estilos-css.html>

[22] Compara tarifas energía:

<http://www.comparatarifasenergia.es/info-energia/mi-factura/electricidad>

[23] Tarifas gasluz:

<http://tarifasgasluz.com/electricidad/tarifa-de-ultimo-recurso-electrica>

[24] Cinco días:

http://cincodias.com/cincodias/2015/06/04/empresas/1433416255_975010.html

Energíadiario:

<http://www.energiadiario.com/publicacion/la-facturacion-electrica-horaria-empezara-a-aplicarse-a-partir-de-octubre/>

EnergyNews:

<http://www.energynews.es/industria-pone-en-marcha-la-nueva-tarifa-de-la-luz-por-horas-para-casi-11-millones-de-hogares/>

[25] REE: www.ree.es

[26] Iberdrola: <http://www.iberdrola.es/consejosdeahorro/>

[27] Stackoverflow:

<http://stackoverflow.com/questions/25685712/connection-to-google-spreadsheets-not-opening-in-shiny-server>

ANEXO 1: ARCHIVO readGoogleSheet

```
readGoogleSheet <- function(url, na.string="", header=TRUE){  
  stopifnot(require(XML))  
  suppressWarnings({  
    #LA SIGUIENTE LÍNEA SE QUITA  
    doc <- paste(readLines(url), collapse=" ")  
    #SE SUSTITUYE POR LA QUE VIENE A CONTINUACIÓN  
    doc <- content(GET(url), as="text")  })  
    if(nchar(doc) == 0) stop("No content found")  
    htmlTable <- gsub("^.*?(<table.*</table>).*$", "\\1>", doc)  
    ret <- readHTMLTable(htmlTable, header=header, stringsAsFactors=FALSE, as.data.frame=TRUE)  
    lapply(ret, function(x){ x[ x == na.string] <- NA; x})  
  }  
}
```

ANEXO 2: ARCHIVO cleanGoogleTable

```

cleanGoogleTable <- function(dat, table=1, skip=0, ncols=NA, nrow=-1,
header=TRUE, dropFirstCol=NA){

  if(!is.data.frame(dat)){

    dat <- dat[[table]]

  }

  if(is.na(dropFirstCol)) {

    firstCol <- na.omit(dat[[1]])

    if(all(firstCol == ".") || all(firstCol==
as.character(seq_along(firstCol)))) {

      dat <- dat[, -1]

    }

  } else if(dropFirstCol) {

    dat <- dat[, -1]

  }

  if(skip > 0){

    dat <- dat[-seq_len(skip), ]

  }

  if(nrow(dat) == 1) return(dat)

  if(nrow(dat) >= 2){

    if(all(is.na(dat[2, ]))) dat <- dat[-2, ]

  }

  #LAS SIGUIENTES LÍNEAS QUE ESTÁN COMENTADAS, SE BORRAN POR LOS PROBLE-
  #MAS GENERADOS CON LOS TÍTULOS DE LAS TABLAS

  #   if(header && nrow(dat) > 1){

  #     header <- as.character(dat[1, ])

  #     names(dat) <- header

  #     dat <- dat[-1, ]

  #   }

  # Keep only desired columns

  if(!is.na(ncols)){

```

```
ncols <- min(ncols, ncol(dat))

dat <- dat[, seq_len(ncols)]

}

# Keep only desired rows

if(nrows > 0){

  nrows <- min(nrows, nrow(dat))

  dat <- dat[seq_len(nrows), ]

}

# Rename rows

rownames(dat) <- seq_len(nrow(dat))

dat

}
```


ANEXO 3: ARCHIVO CSS

```
h1 {  
    text-align: center;  
    color: #1E90FF; text-transform: uppercase;  
    font-family: oblique bold 120% cursive;  
  
    font-weight: 500;  
    line-height: 1.1;  
  
}  
  
h3{  
    color: #3137B8  
}  
  
h4{  
    color: #656ACC  
}  
  
h5{  
    font-weight: 400  
}  
  
body {  
    background-color: #fff;  
}
```

ANEXO 4: SCRIPT ui.R

```
library(shiny)

library(RCurl)      #Para subir archivos de google drive

library(repmis)

library(bitops)

library(httr)

library(gsheet)

library(data.table)

# Define UI

shinyUI(fluidPage(

  includeCSS("estilo.css"),

  #Título e imagenes

tags$div(

  HTML("</head>

  <body>

  <div class='container-fluid'>

  <div class='row-fluid'>

  <div class='span2'>

  </div>

  <div class='span8'>

  </div>

  <div class='span2'>

  <img align=center src= 'http://oi57.tinypic.com/r224jr.jpg' />

  ")),
```

pestañas

```

navbarPage("",
    tabPanel("1- Consumo diario",
        sidebarPanel(
            wellPanel(
                h3(strong("Resumen de datos: ")),
                br(),
                h4(strong("Día actual")),
                h4("·Fecha y hora de la consulta: "),
                h5(textOutput('fecha_hora')),
                h4("·Consumo instantaneo de potencia (W):
                "),
                h5(textOutput('dato_diezminutal')),
                h4("·Consumo diario de energía (kWh): "),
                h5(textOutput('dato_diario')),
                br(),
                h4(strong("Día anterior")),
                h4("·Fecha del día de ayer: "),
                h5(textOutput('dia_ayer')),
                h4("·Consumo de energía (kWh): "),
                h5(textOutput('dato_dia_anterior'))

            )),
        mainPanel(
            plotOutput('grafica_comparativa'),
            h3(textOutput('aviso_no_dia_anterior')),

```

```

        h3(textOutput('aviso_consumo_diario')),
        plotOutput('grafica_ultimas24h')
    ) ),
    tabPanel("2- Consumo mensual",
        sidebarPanel(
            wellPanel(
                h3(strong("Resumen de datos: ")),
                h4(strong("Mes: ")),
                h5(strong(textOutput('nombre_mes'))),
                br(),
                h4(strong("Año actual")),
                h4("·Año: "),
                h5(textOutput('año1')),
                h4("·Consumo mensual de energía (kWh): "),
                h5(textOutput('dato_mensual')),
                br(),
                h4(strong("Año pasado")),
                h4("·Año: "),
                h5(textOutput('año_anterior1')),
                h4("·Consumo mensual de energía (kWh): "),
                h5(textOutput('dato_mes_año_pasado'))
            ),
            mainPanel(
                plotOutput('grafica_comparativa2'),
                h3(textOutput('aviso_consumo_mensual'))
            )
        ) ),

```

```

tabPanel("3- Consumo anual",
  sidebarPanel(
    wellPanel(
      h3(strong("Resumen de datos: ")),
      br(),
      h4(strong("Año actual")),
      h4("·Año: "),
      h5(textOutput('año')),
      h4("·Consumo anual de energía (kWh): "),
      h5(textOutput('dato_anual')),
      br(),
      h4(strong("Año pasado")),
      h4("·Año: "),
      h5(textOutput('año_anterior')),
      h4("·Consumo anual de energía (kWh): "),
      h5(textOutput('consumo_año_anterior'))
    ),
    mainPanel(
      plotOutput('grafica_comparativa3'),
      h3(textOutput('aviso_3'))
    )
  ),
tabPanel("4- Datos significativos",
  navbarPage("",

    tabPanel("Mes del año de mayor consumo",
      sidebarPanel(
        wellPanel(
          h3(strong("Mes del año

```

```

        de mayor consumo de
        energía")),

    br(),

    h4(strong("·Mes: ")),

    h5(textOutput("nombre_mes_max")),

    h4(strong("·Consumo de
    energía (KWh): ")),

    h5(textOutput('consumo_mes_max')),

    h4(strong("·Factura (Euros) : ")),

    h5(textOutput('gasto_mes_max'))

    )),

mainPanel(

    h3(textOutput("aviso_max")),

    plotOutput('grafica_mes_max')

    )),

tabPanel("Mes del año de menor consumo",

    sidebarPanel(

        wellPanel(

            h3(strong("Mes del año
            de menor consumo de
            energía: ")),

            h5("(No se tiene en
            cuenta el mes actual
            para realizar el cálculo)"),

```

```

        br(),
        h4(strong("·Mes: ")),

        h5(textOutput('nombre_mes_min')),

        h4(strong("·Consumo de energía (KWh): ")),

        h5(textOutput('consumo_mes_min')),

        h4(strong("·Factura (Euros) : ")),

        h5(textOutput('gasto_mes_min'))
    )),
    mainPanel(

        h3(textOutput("aviso_min")),

        plotOutput('grafica_mes_min')
    )),
    tabPanel("Día del año de mayor consumo",
        sidebarPanel(
            wellPanel(

                h3(strong("Día del año de mayor consumo de energía")),

                br(),

                h4(strong("·Dia:")),

                h5(textOutput('dia_max')),

                h4(strong("·Consumo de energía (kWh): ")),
            )
        )
    )

```

```

        h5(textOutput('valor_di
a_max_consumo'))

   )),

mainPanel(

    plotOut-
put('grafica_dia_max')

)

),

tabPanel("Día del año de menor consumo",

    sidebarPanel(

        wellPanel(

            h3(strong("Día del año
de menor consumo de
energía")),

            h5("No se tiene en
cuenta el día de hoy pa-
ra realizar el cálcu-
lo")),

            br(),

            h4(strong("·Dia:")),

            h5(textOutput('dia_min'
)),

            h4(strong("·Consumo de
energía (kWh): ")),

            h5(textOutput('valor_dia
_menor_consumo'))

        )),

    mainPanel(

        plotOut-
put('grafica_dia_min')
    )
)

```



```

        )
    ),
    id="pestanas")
),

tabPanel("5-Factura eléctrica",
    navbarPage("",
        tabPanel("Factura eléctrica del mes actual",
            sidebarPanel(
                wellPanel(
                    h3(strong("Resumen de
datos: ")),

                    h4(strong(textOutput('a
ño2'))),

                    h4(strong(textOutput('n
ombre_mes1'))),

                    h4(strong(textOutput('c
ons1'))),

                    br(),

                    h3(strong("Detalles del
contrato: ")),

                    h4(strong("·Potencia
contratada: 3.3 kW")),

                    h4(strong("·Precio de la
potencia contratada:
0.115383 Euros/kW ")),

                    h4(strong("·Precio kwh:
kWh ")),

                    br()

                )),
    ),

```

```
mainPanel(  
  
    h3(strong("Detalle factu-  
ra")),  
  
    h4(strong("Precio de la  
potencia contratada (Eu-  
ros): ")),  
  
    h5(textOutput('precio_pote  
ncial')),  
  
    h4(strong("Consumo (Eu-  
ros): ")),  
  
    h5(textOutput('precio_cons  
1')),  
  
    h4(strong("Impuesto elec-  
tricidad (Euros):")),  
  
    h5(textOutput('impto_elect  
ricidad1')),  
  
    h4(strong("Servicio pro-  
tección de pagos (Eu-  
ros):")),  
  
    h5(textOutput('servicio_pr  
oteccion1')),  
  
    h4(strong("Alquiler de  
equipos (Euros): ")),  
  
    h5(textOutput('alquiler_eq  
uipos1')),  
  
    h4(strong("TOTAL (Euros):  
")),  
  
    h5(textOutput('total1')),  
  
    h4(strong("IVA (21%) :")),  
  
    h5(textOutput('precio_i  
val')),  
  
    h4(strong("FACTURA TOTAL:  
")),
```

```
h5(strong(textOutput('factura_total1'))),  
  
br(),  
  
plotOutput('grafica_consumo_12meses')  
  
)),  
  
tabPanel("Factura eléctrica  
del mes anterior",  
  
sidebarPanel(  
  
  wellPanel(  
  
    h3(strong("Resumen de  
datos: ")),  
  
    h4(strong(textOutput('año3'))),  
  
    h4(strong(textOutput('nombre_mes2'))),  
  
    h4(strong(textOutput('consumos2'))),  
  
    br(),  
  
    h3(strong("Detalles del  
contrato:")),  
  
    h4(strong("·Potencia  
contratada: 3.3 kW")),  
  
    h4(strong("·Precio de la  
potencia contratada:  
0.115383 Euros/kW ")),  
  
    h4(strong("·Precio kWh:  
0.129862 Euros/kWh ")),
```

```
br()

)),

mainPanel(

    h3(textOutput('aviso_fa
ctura')),

    h3(strong("Detalle fac-
tura")),

    h4(strong("Precio de la
potencia contratada
(Euros): ")),

    h5(textOutput('precio_p
otencia2')),

    h4(strong("Consumo (Eu-
ros): ")),

    h5(textOutput('precio_c
ons2')),

    h4(strong("Impuesto
electricidad (Eu-
ros):")),

    h5(textOutput('impto_el
ectricidad2')),

    h4(strong("Servicio
protección de pagos
(Euros):")),

    h5(textOutput('servicio
_proteccion2')),

    h4(strong("Alquiler de
equipos (Euros): ")),

    h5(textOutput('alquiler
_equipos2')),

    h4(strong("TOTAL (Eu-
ros): ")),
```

```
h5(textOutput('total2')),  
h4(strong("IVA (21%) :")),  
  
h5(textOutput('precio_iva  
2')),  
h4(strong("FACTURA TOTAL:  
")),  
  
h5(strong(textOutput('fact  
ura_total2'))),  
br(),  
  
plotOut-  
put('grafica_consumo_12mese  
s2')  
  
),  
  
id="pestana")  
  
,  
tabPanel("6- Consejos",  
  navbarPage("",  
    tabPanel("Ahorro de luz",  
  
      mainPanel(  
  
        h1("CONSEJOS"),  
  
        br(),  
  
        h2("Disminución del consumo de la luz"),
```

```

br() ,

p("1.- Colocar interruptores divididos para poder encender únicamente las luces que sean necesarias."),

p("2.- Cambiar las bombillas tradicionales, por bombillas de bajo consumo, que ahorran hasta el 80% de energía eléctrica y duran 8 veces más."),

p("3.- Aprovechar la iluminación natural."),

p("4.- Apagar las luces de las habitaciones que no estés utilizando."),

p("5.- Las lámparas halógenas proporcionan más luz y su vida es de dos a tres veces mayor."),

p("6.- En las estancias donde la luz debe estar encendida durante más tiempo, el uso de tubos fluorescentes permite un ahorro de energía de hasta el 80%, y duran de 8 a 10 veces más que las bombillas incandescentes. Las fluorescentes consumen más energía en el encendido, por lo que hay que tener en cuenta que no se estén encendiendo y apagando continuamente."),

p("7.- Si dispones de iluminación exterior en tu jardín, conviene controlar su funcionamiento mediante un programador o interruptor crepuscular."),

p("8.- Se deben utilizar colores claros en paredes y techos para aprovechar mejor la iluminación natural y reducir la artificial."),

p("9.- La colocación de reguladores de intensidad luminosa de tipo electrónico ayudan al ahorro de energía. "),

p("10.- Se ahorra energía con detectores de presencia en garajes y zonas comunes para que las luces funcionen automáticamente."),

p("11.- La iluminación localizada y la adaptación de la misma a las necesidades hace que se ahorre energía y se consigan ambientes más confortables."),

br() ,

HTML("<img
src='http://www.isotools.org/wp-content/uploads/2012/12/bombilla-idea-innovacion-tecnologia640.jpg'/>")

)
    
```

```

    ),
    tabPanel("Ahorro en la cocina",
        mainPanel(
            h1("CONSEJOS"),
            br(),
            h2("Disminución del consu-
mo en la cocina"),
            br(),
            p("1.- Al cocinar, se
aprovecha el calor residual si se apaga la cocina unos minutos antes
de retirar el recipiente. (salvo para cocinas de inducción)"),
            p("2.- Usar un recipiente
con un diámetro ligeramente mayor que la zona de cocción y con fondo
difusor de calor (salvo para cocinas de inducción), supone un ahorro
en la energía consumida para cocinar del 20%.") ,
            p("3.- El uso de la olla a
presión disminuye la energía consumida y se ahorra mucho tiempo."),
            p("4.- Conviene regular la
temperatura del frigorífico y del congelador según las instrucciones
del fabricante."),
            p("5.- El frigorífico y el
congelador se deben instalar lo más lejos posible de los focos de ca-
lor."),
            p("6.- No se deben intro-
ducir alimentos calientes en el frigorífico, se deben dejar enfriar
antes de introducirlos en el mismo."),
            p("7.- Una buena elección
a la hora de comprar el frigorífico es determinante, conviene elegir
modelos eficientes y ecológicos."),
            p("8.- Es importante eli-
minar la capa de hielo que se crea en el congelador, ya que así se
consiguen ahorros del 30% de la energía."),
            p("9.- No se debe abrir el
horno innecesariamente ya que se puede perder hasta un 20% del calor
acumulado."),
            p("10.- Para poder contro-
lar la pérdida de energía es importante usar un reloj programador en
el horno."),

```

```

        p("11.- Es importante
aprovechar al máximo la capacidad del horno, y esto se consigue coci-
nando el mayor número de alimentos a la vez. "),

        p("12.- Si se va a cocinar
durante más de una hora no es necesario precalentar el horno."),

        p("13.- Se debe aprovechar
el calor residual del horno, por ello, se debe apagar el horno un poco
antes de finalizar la cocción."),

        p("14.- Es interesante el
uso de los hornos de convección, ya que favorecen la distribución uni-
forme de calor con su consiguiente ahorro de tiempo."),

        p("15.- El uso del micro-
ondas supone un ahorro energético de un 60-70% respecto de los hornos
convencionales."),

        p("16.- Con el uso del la-
vavajillas se consume un 40% menos de energía que con agua calien-
te."),

        p("17.- Es importante el
uso adecuado de un programa determinado para cada lavado con el lava-
vajillas y llenar el mismo en cada ocasión."),

        p("18.- Se deben colocar
los utensilios de una forma adecuada para que el agua circule con fa-
cilidad."),

        p("19.- Usa los niveles de
sal y abrillantador recomendados por el fabricante de lavavajillas."),

        p("20.-Compra un lavavaji-
llas eficiente y ecológico."),

        br(),

        HTML("<img
src='http://www.ifeelmaps.com/system/cms/files/files/000/000/401/origi-
nal/cocinar_ecolo_23.jpg'/>")

    )

),

tabPanel("Electrodomésticos",

    mainPanel(

        h1("CONSEJOS"),

        br(),

        h2("Disminución de consumo
debido al buen uso de los electrodomésticos."),
    
```


br()),

p("1.- El uso de una temperatura moderada en la lavadora reduce mucho el gasto energético."),

p("2.- Aprovechar al máximo la carga de la lavadora o secadora reduce el consumo energético y alarga la vida útil del electrodoméstico."),

p("3.- Al igual que con el frigorífico y el lavavajillas, a la hora de comprar una nueva lavadora se deben escoger modelos eficientes y ecológicos."),

p("4.- Un centrifugado de la lavadora a 1200 revoluciones, en vez de a 700 reduce el consumo de la secadora hasta un 20%."),

p("5.- Elige ordenadores, impresoras u otros electrodomésticos con ahorro de energía."),

p("6.- Ayudan en el ahorro de energía las lavadoras: termoeeficientes, con programas de media carga y con sonda de agua."),

p("7.- Conviene aprovechar el calor del sol para secar la ropa."),

p("8.- El uso de descalcificantes y una limpieza regular del filtro de la lavadora ayuda en ahorro de energía."),

p("9.- Si tienes contratada la Tarifa con discriminación horaria, aprovecha las horas de descuento para poner la lavadora."),

p("10.- Plancha grandes cantidades de ropa de una vez para aprovechar el calentamiento de la plancha."),

p("11.- Optimiza el uso de los aparatos eléctricos."),

p("12.- Apaga del todo (desenchufa) totalmente los televisores y equipos en ausencia superiores a 30 minutos."),

p("13.- Apaga la pantalla del ordenador cuando no vayas a utilizarlo durante periodos cortos."),

p("14.- El salvapantallas que menos consume es el de color negro."),

p("15.- Los ladrones de conexión múltiple ayudan en el ahorro de energía."),

```
p("16.- Un buen manteni-  
miento y la limpieza de los electrodomésticos prolonga su vida útil y  
ahorra energía."),
```

```
br(),
```

```
HTML("<img  
src='http://www.certificadoenergeticoburgos.org/images/fotolia_4691844  
8_s-copia-940x400.jpg'/>")
```

```
)
```

```
),
```

```
id="pestana")
```

```
),
```

```
id="pestana")
```

```
))
```

ANEXO 5: SRIPT server.R

```
library(shiny)

library(RCurl)

library(XML)

library(data.table)

library(stringi)

library(xtable)

library(bitops)

readGoogleSheet <- function(url, na.string="", header=TRUE){

  stopifnot(require(XML))

  # Suppress warnings because Google docs seems to have incomplete fi-
  nal line

  suppressWarnings({

    doc <- content(GET(url), as="text" ) )

    if(nchar(doc) == 0) stop("No content found")

    htmlTable <- gsub("^.*?(<table.*</table>).*$", "\\1>", doc)

    ret <- readHTMLTable(htmlTable, header=header, stringsAsFac-
    tors=FALSE, as.data.frame=TRUE)

    lapply(ret, function(x){ x[ x == na.string] <- NA; x})

  })

cleanGoogleTable <- function(dat, table=1, skip=0, ncols=NA, nrows=-1,
header=TRUE, dropFirstCol=NA){

  if(!is.data.frame(dat)){

    dat <- dat[[table]]

  }

  if(is.na(dropFirstCol)) {

    firstCol <- na.omit(dat[[1]])
```

```
  if(all(firstCol == ".") || all(firstCol==
as.character(seq_along(firstCol)))) {

  dat <- dat[, -1]

}

} else if(dropFirstCol) {

  dat <- dat[, -1]

}

if(skip > 0){

  dat <- dat[-seq_len(skip), ]

}

if(nrow(dat) == 1) return(dat)

if(nrow(dat) >= 2){

  if(all(is.na(dat[2, ]))) dat <- dat[-2, ]

}

#   if(header && nrow(dat) > 1){

#     header <- as.character(dat[1, ])

#     names(dat) <- header

#     dat <- dat[-1, ]

#   }

# Keep only desired columns

if(!is.na(ncols)){

  ncols <- min(ncols, ncol(dat))

  dat <- dat[, seq_len(ncols)]

}

# Keep only desired rows

if(nrows > 0){

  nrows <- min(nrows, nrow(dat))

  dat <- dat[seq_len(nrows), ]

}

# Rename rows

rownames(dat) <- seq_len(nrow(dat))
```

```

    dat
}

#cargar las funciones para poder leer los archivos de google drive

shinyServer(function(input,output,session) {

    # CONSTANTES PARA LA FACTURA

    precio_electricidad=0.129862    #€/kWh

    iva=0.21

    potencia_contratada=3.3        #3.3kW

    precio_potencia=0.115383      #€/kWh

    impuesto_electricidad=1.05113*0.04864    #impuesto de electricidad:
    1.05113*4.864%

    serv_prot=0.95                #servicio protección de pagos 0.95
    euros/mes

    alquilerEquipos=0.58

    #cargar dato de último dato diezminutal registrado

    datos_potencia_diezminutal <-reactive({

        invalidateLater(300000,session)

        gdoc <-
        "http://docs.google.com/spreadsheets/d/18T0kZWzCrncctjeVuXygUF1MHkzfUX
        4QU7Tm1D4qhNY/gviz/tq?tqx=out:html&tq=SELECT%20year(A)%20month(A)%20Cda
        y(A)%20Chour(A)%20Cminute(A)%20Csecond(A)%20C(B*230*0.9)%20ORDER%20BY%20A%
        20DESC%20LIMIT%201 "

        elem <- readGoogleSheet(gdoc)

        m <- cleanGoogleTable(elem, table=1)

    })
}

```

```
#cargar datos horarios (365*24 datos al año)

datos_horarios<-reactive({

  invalidateLater(30000,session)

  gdoc <-
"http://docs.google.com/spreadsheets/d/18T0kZWzCrncctjeVuXygUF1MHkzfUX
4QU7TmlD4qhNY/gviz/tq?tqx=out:html&tq=select%20year(A)%20month(A)%20%2
0day(A)%20%20hour(A)%20%230*0.9*(sum(B))%20F6%20group%20by%20year(A)%20Cm
onth(A)%20%20day(A)%20%20hour(A)"

  elem <- readGoogleSheet(gdoc)

  m <- cleanGoogleTable(elem, table=1)

})

#cargar datos de consumo de energía diaria(365 datos para cada
año),cargar y escribir ultimo día

datos_diarios <-reactive({

  invalidateLater(300000,session)

  gdoc <-
"http://docs.google.com/spreadsheets/d/18T0kZWzCrncctjeVuXygUF1MHkzfUX
4QU7TmlD4qhNY/gviz/tq?tqx=out:html&tq=select%20year(A)%20month(A)%20Cda
y(A)%20%230*0.9*(sum(B))%20F6%20group%20by%20year(A)%20month(A)%20Cday(A)
"

  elem <- readGoogleSheet(gdoc)

  m <- cleanGoogleTable(elem, table=1)

})

#Cargar datos de enegía mensual consumida (12 datos de cada año, es
decir, datos mensuales),tabla y escribir último mes

datos_mensuales <-reactive({

  invalidateLater(300000,session)

  gdoc <-
"http://docs.google.com/spreadsheets/d/18T0kZWzCrncctjeVuXygUF1MHkzfUX
4QU7TmlD4qhNY/gviz/tq?tqx=out:html&tq=select%20year(A)%20month(A)%20C23
0*0.9*(sum(B))%20F6%20group%20by%20year(A)%20month(A)"

  elem <- readGoogleSheet(gdoc)

  m <- cleanGoogleTable(elem, table=1)
```

```
  })

  #Cargar consumo anual de energía (1 unico dato de cada año: computo
de energía anual consumida)

  datos_anuales <-reactive({

    invalidateLater(300000,session)

    gdoc <-
"http://docs.google.com/spreadsheets/d/18T0kZWzCrncctjeVuXygUF1MHkzfUX
4QU7TmlD4qhNY/gviz/tq?tqx=out:html&tq=select%20year(A)%2C230*0.9*(sum(
B))%2F6%20group%20by%20year(A) "

    elem <- readGoogleSheet(gdoc)

    m <- cleanGoogleTable(elem, table=1)

  })

#calculo de numero de horas, dias, meses y años

numero_horas<-reactive({

  filas<-nrow(datos_horarios())

})

numero_dias<-reactive({

  filas<-nrow(datos_diarios())

})

numero_meses<-reactive({

  filas<-nrow(datos_mensuales())

})

numero_años<-reactive({

  filas<-nrow(datos_anuales())

})
```

```
#####CONSUMO DIARIO#####
```

```
#mostrar ultimo dato diezminutal con la fecha

output$dato_diezminutal<-renderText({
  aux<-datos_potencia_diezminutal()[1,7]
  a<-as.numeric(as.character(aux))
  b<-round(a,2)
  paste(b,"W")
})

output$fecha_hora<-renderText({
  año<-datos_potencia_diezminutal()[1,1]
  aux<-datos_potencia_diezminutal()[1,2]
  mes<-as.numeric(as.character(aux))

  nombre_mes<-
switch(mes+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio",
        "Ju-
lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

  dia<-datos_potencia_diezminutal()[1,3]
  hora<-datos_potencia_diezminutal()[1,4]
  if (hora<10){
    horal=paste("0",hora)
  }
  else(horal=hora)
  minutos<-datos_potencia_diezminutal()[1,5]
  if (minutos<10){
    minutos1=paste("0",minutos)
  }
  else(minutos1=minutos)

  fecha_hoy<-paste(dia,"de",nombre_mes,"del",año, "
",horal,":",minutos1)
})
```



```
#salida de dato en kw

output$dato_diario<-renderText({

  aux<-datos_diarios()[numero_dias(),4]

  a<-as.numeric(as.character(aux))

  b<-round(a/1000,2)

  paste(b,"kWh")

})

#dato del consumo del día anterior

output$dato_dia_anterior<-renderText({

  if (numero_dias()==1){

    c<-paste("No hay información sobre día anterior")

  }

  if (numero_dias(>1){

    aux<-datos_diarios()[ (numero_dias()-1),4]

    a<-as.numeric(as.character(aux))

    b<-round(a/1000,2)

    paste(b,"kWh")

  }

})

output$dia_ayer<-renderText({

  if (numero_dias()==0){

    c<-paste("No hay información sobre día anterior")

  }

  if (numero_dias(>1){
```

```

    dia<-datos_diarios()[(numero_dias()-1),3]

    aux<-datos_diarios()[(numero_dias()-1),2]

    mes<-as.numeric(as.character(aux))

    nombre_mes<-
switch(mes+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    año<-datos_diarios()[(numero_dias()-1),1]

    fecha_ayer<-paste(dia,"de",nombre_mes,"del",año)

}

}))

#comparativa de los dos últimos días
output$grafica_comparativa1<-renderPlot({
    aux<-datos_horarios()[(numero_horas()-4),4]

    hora<-as.numeric(as.character(aux))

    b<- datos_horarios()[(numero_horas()-hora):(numero_horas()),5]

    y<-as.numeric(as.character(b))/1000

    if (numero_dias()<3) {

        z<-datos_horarios()[(numero_horas()-hora):(numero_horas()),4]

        nombre_hora<-as.numeric(as.character(z))

        limite=max(y)+0.6

        barplot(y,col="blue",width=1,xlab="horas",ylab="Energía consumi-
da (kWh)",border="blue",

                ylim=c(0,limite),xlim=c(0,28),names.arg=nombre_hora)

        title(main="Consumo del día actual", col.main="blue",
font.main=4)

    }

    if (numero_dias()>=3){

        c<-datos_horarios()[(numero_horas()-hora-24):(numero_horas()-
hora-1),5]
    }

```

```

y_anterior<-as.numeric(as.character(c))/1000

y_ant_max<-max(y_anterior)

y_max<-max(y)

if ((y_ant_max)>(y_max)){
  limite=y_ant_max+0.6
}
else limite=y_max+0.6

barplot(y,col="blue",width=1,xlab="horas",ylab="Energía consumi-
da (kWh)",border="blue",

        ylim=c(0,limite),xlim=c(0,28))

par(new=TRUE)

barplot(y_anterior,col="black",width=1,axes=FALSE,density=40,xlim=c(0,
28),

        ylim=c(0,limite),names.arg=c("0h"," ","2h"," ","4h","
","6h"," ","8h"," ","10h"," ","12h",

                                     " ","14h"," ","16h","
","18h"," ","20h"," ","22h"," "))

title(main="Comparativa del consumo con el día anterior",
col.main="blue", font.main=4)

legend("topleft", c("Consumo hoy", "Consumo
ayer"),fill=c("blue","black"))

}

})

#gráfica horaria de las últimas 24h

output$grafica_ultimas24h<- renderPlot({

  if (numero_horas()<24){

    a<-datos_horarios()[1:(numero_horas()),5]
  }
})

```

```

y<-as.numeric(as.character(a))/1000

b<-datos_horarios()[1:(numero_horas()),4]

nombre_horas<-as.numeric(as.character(b))

y_max<-max(y)

limite=y_max+0.1

barplot(y,col="blue",xlab="horas",ylab="Energía consumida
(kWh)",border="blue",ylim=c(0,limite),xlim=c(0,28),names.arg=nombre_ho
ras)

title(main="Consumo de las últimas 24h", col.main="blue",
font.main=4)

}

if (numero_horas())>=24){

a<-datos_horarios()[ (numero_horas()-23):(numero_horas()),5]

y<-as.numeric(as.character(a))/1000

b<-datos_horarios()[ (numero_horas()-23):(numero_horas()),4]

c<-c(paste(b[1],"h")," ",paste(b[3],"h")," ",paste(b[5],"h"),"
",paste(b[7],"h")," ",paste(b[9],"h")," ",paste(b[11],"h"),"
",paste(b[13],"h")," ",paste(b[15],"h")," ",paste(b[17],"h"),"
",paste(b[19],"h")," ",paste(b[21],"h")," ",paste(b[23],"h")," ")

y_max<-max(y)

limite=y_max+0.1

barplot(y,col="blue",xlab="horas",ylab="Energía consumida
(kWh)",border="blue",ylim=c(0,limite),xlim=c(0,28),names.arg=c)

title(main="Consumo de las últimas 24h", col.main="blue",
font.main=4)

}

})

#aviso

```

```
output$aviso_consumo_diario<-renderText({
  a<-datos_diarios()[numero_dias()-1,4]
  consumo_ayer<-as.numeric(as.character(a))
  b<-datos_diarios()[numero_dias(),4]
  consumo_hoy<-as.numeric(as.character(b))
  if (numero_horas())>48){
    if ((consumo_hoy)>(0.8*(consumo_ayer))){
      texto<-paste("Has consumido más del 80% de la energía que consumiste el día de ayer. Accede a la pestaña de consejos para aprender a reducir el consumo.¡Reduce el consumo!")
    }
  }
})
```

```
output$aviso_no_dia_anterior<-renderText({
  if (numero_horas())<48){
    paste("No hay 24 datos del día anterior, por lo que no se muestra la comparación entre el consumo del día de hoy y el día de ayer.")
  }
})
```

#####CONSUMO MENSUAL#####

#datos de consumo mensual

```
output$dato_mensual<-renderText({
  aux<-datos_mensuales()[numero_meses(),3]
  a<-as.numeric(as.character(aux))
  b<-round(a/1000,2)
  paste(b,"kWh")
})
```

```
  })

  output$nombre_mes<-renderText ({
    aux<-datos_mensuales () [numero_meses (),2]
    mes<-as.numeric (as.character (aux))

    switch (mes+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
    "Agosto","Septiembre","Octubre","Noviembre","Diciembre")

  })

  output$año1<-renderText ({
    año<-datos_mensuales () [numero_meses (),1]

  })

  output$año_anterior1<-renderText ({
    if (numero_años ()==1){
      aux<-paste ("No hay datos del año anterior")
    }
    else
      aux<-datos_anuales () [numero_años ()-1,1]

  })

  output$dato_mes_año_pasado<-renderText ({
    if (numero_años ()==1){
      aux<-paste ("No hay datos del año anterior")}
    else
      aux<-datos_mensuales () [numero_meses ()-12,3]
```

```
a<-as.numeric(as.character(aux))

b<-round(a/1000,2)

paste(b, "kWh")

}))

#gráfica comparativa con el mes del año pasado

#numero de días de cada año (calculo de numero de veces que aparece
cada año en la tabla)

tabla_numero_dias_cada_año<-reactive({
  zaux<-datos_diarios()[1:numero_dias()],1]
  vector_años<-as.numeric(as.character(zaux))
  tabla_frecuencias<-table(vector_años)
})

numero_dias_año_pasado<-reactive({
  filas<-nrow(tabla_numero_dias_cada_año())
  dato<-tabla_numero_dias_cada_año()[filas-1]
})

#calculo del numero de días que tienen los ultimos doce meses. Usa-
mos la funcion de calculo de frecuencias(table). Estan ordenados desde
enero hasta diciembre.

tabla_numero_dias_ultimos_12_meses<-reactive({
  aux<-datos_diarios()[numero_dias(),3]
```

```

dias<-as.numeric(as.character(aux))

if (numero_años(>1){

    zaux<-datos_diarios()[((numero_dias()-dias-
numero_dias_año_pasado()): (numero_dias()-dias)),2]

    vector_meses<-as.numeric(as.character(zaux))

    tabla_frecuencias<-table(vector_meses)

}

else{

    if (numero_dias(>dias){

        zaux<-datos_diarios()[1:(numero_dias()-dias),2]

        vector_meses<-as.numeric(as.character(zaux))

        tabla_frecuencias<-table(vector_meses)

    }

    if (numero_dias(<=dias){

        tabla_frecuencias=0

    }

}

})

#acceso numero de dias que tenia el mes actual el año pasado

dias_mes_año_pasado<-reactive({

    aux<-datos_mensuales()[numero_meses(),2]

    mes<-as.numeric(as.character(aux))

    dato<-tabla_numero_dias_ultimos_12_meses()[mes+1]

})

output$grafica_comparativa2<-renderPlot({

```



```

if (numero_meses()>12) {
  aux<-datos_diarios()[numero_dias(),3]
  dias<-as.numeric(as.character(aux))
  b<-datos_diarios()[ (numero_dias()-dias+1):(numero_dias()),4]
  y<-as.numeric(as.character(b))/1000
  c<-datos_diarios()[ (numero_dias()-dias-
numero_dias_año_pasado()+1):(numero_dias()-dias-
numero_dias_año_pasado()+dias_mes_año_pasado()),4]
  y_anterior<-as.numeric(as.character(c))/1000
  d<-datos_diarios()[ (numero_dias()-dias-
numero_dias_año_pasado()+1):(numero_dias()-dias-
numero_dias_año_pasado()+dias_mes_año_pasado()),3]
  eje_x<-as.numeric(as.character(d))
  y_ant_max<-max(y_anterior)
  y_max<-max(y)
  if ((y_ant_max)>(y_max)){
    limite=y_ant_max+8
  }
  else limite=y_max+8
  barplot(y,col="blue",width=1,xlab="días",ylab="Energía consumida
(kWh)",border="blue",xlim=c(0,40),ylim=c(0,limite))
  par(new=TRUE)
  barplot(y_anterior,col="black",width=1,axes=FALSE,density=40,xlim=c(0,
40),ylim=c(0,limite),names.arg=eje_x)
  title(main="Comparativa del consumo anual",col.main="blue",
font.main=4)
  legend("topleft", c("Mes del año actual", "Mes del año pasa-
do"),fill=c("blue","black"))
}

if (numero_meses()<=12){
  aux<-datos_diarios()[numero_dias(),3]

```

```

dias<-as.numeric(as.character(aux))

if (numero_dias()>dias){

  b<-datos_diarios()[(numero_dias()-dias+1):(numero_dias()),4]

  y<-as.numeric(as.character(b))/1000

  limite<-max(y)+8

  v<-datos_diarios()[(numero_dias()-dias+1):(numero_dias()),3]

  eje_x<-as.numeric(as.character(v))

  barplot(y,col="blue",width=1,xlab="días",ylab="Energía consu-
mida
(kWh)",border="blue",xlim=c(0,40),ylim=c(0,limite),names.arg=eje_x)

  title(main="Consumo mensual",col.main="blue", font.main=4)

}

if (numero_dias()<=dias){

  b<-datos_diarios()[1:(numero_dias()),4]

  y<-as.numeric(as.character(b))/1000

  limite<-max(y)+5

  v<-datos_diarios()[1:(numero_dias()),3]

  eje_x<-as.numeric(as.character(v))

  barplot(y,col="blue",width=1,xlab="días",ylab="Energía consu-
mida
(kWh)",border="blue",xlim=c(0,40),ylim=c(0,limite),names.arg=eje_x)

  title(main="Consumo mensual",col.main="blue", font.main=4)

}

}

}))

```

```

output$aviso_consumo_mensual<-renderText({

  a<-datos_mensuales()[numero_meses(),3]

  consumo_mes_actual<-as.numeric(as.character(a))

  b<-datos_mensuales()[numero_meses()-12,3]

```

```

consumo_mes_pasado<-as.numeric(as.character(b))

if (numero_meses()>12){

  if ((consumo_mes_actual)>(0.8*(consumo_mes_pasado))){

    texto<-paste("Has consumido más del 80% de la energía que con-
sumiste este mismo mes el año pasado. Accede a la pestaña de consejos
para aprender a reducir el consumo.¡Reduce el consumo!")

  }

}

if (numero_meses()<=12){

  texto<-paste("No hay datos del año anterior, por lo que no se
puede comparar con el mes del año pasado")

}

})

```

#####CONSUMO ANUAL#####

```

numero_años<-reactive({

  filas<-nrow(datos_anuales())

})

```

```

output$año<-renderText ({

  aux<-datos_anuales()[numero_años(),1]

})

```

```

output$año_anterior<-renderText ({

  if (numero_años()==1){

    aux<-paste("No hay datos del año anterior")

  }

}

```

```
else
  aux<-datos_anuales() [numero_años()-1,1]

})

output$dato_anual<-renderText({
  aux<-datos_anuales() [numero_años(),2]
  a<-as.numeric(as.character(aux))
  b<-round(a/1000,2)
  paste(b,"kWh")
})

output$consumo_año_anterior<-renderText({
  if (numero_años()==1){
    c<-paste("No hay datos del año anterior")
  }
  if (numero_años(>1){
    aux<-datos_anuales() [(numero_años()-1),2]
    a<-as.numeric(as.character(aux))
    b<-round(a/1000,2)
    c<-paste(b,"kWh")
  }
})
```

```
#gráfica comparativa meses de este año con el año anterior

output$grafica_comparativa3<-renderPlot({
  aux<-datos_mensuales()[numero_meses(),2]
  meses<-as.numeric(as.character(aux))

  if (numero_meses()>12){
    b<-datos_mensuales()[ (numero_meses()-meses):(numero_meses()),3]
    y<-as.numeric(as.character(b))/1000
    c<-datos_mensuales()[ (numero_meses()-meses-12):(numero_meses()-meses-1),3]
    y_anterior<-as.numeric(as.character(c))/1000
    y_max<-max(y)
    y_ant_max<-max(y_anterior)
    if ((y_ant_max)>(y_max)){
      limite=y_ant_max+70
    }
    else limite=y_max+70

    barplot(y,col="blue",width=1,xlab="meses",ylab="Energía consumida (kWh)",border="blue",ylim=c(0,limite),xlim=c(0,17))

    par(new=TRUE)

    barplot(y_anterior,col="black",width=1,axes=FALSE,density=40,xlim=c(0,17),ylim=c(0,limite),names.arg=c("Ener","Febr","Marz","Abril","Mayo","Junio","Julio","Agos","Sept","Oct","Novie","Dicie"))

    title(main="Comparativa del consumo anual", col.main="blue", font.main=4)

    legend("topleft", c("Año actual", "Año anterior"),fill=c("blue","black"))

  }
}
```

```

if (numero_meses() <= 12) {
  b <- datos_mensuales()[1:(numero_meses()), 3]
  y <- as.numeric(as.character(b)) / 1000
  limite <- max(y) + 70
  c <- datos_mensuales()[1:(numero_meses()), 2]
  meses <- as.numeric(as.character(c))
  if (numero_meses() == meses) {
    e <-
    c("Ene", "Febr", "Marz", "Abril", "Mayo", "Junio", "Julio", "Agos", "Sept", "O
    ct", "Novie", "Dicie")
    p <- e[1:meses]
    barplot(y, col="blue", width=1, xlab="meses", ylab="Energía consu-
    mida (kWh)", border="blue", ylim=c(0, limite), xlim=c(0, 17))
    title(main="Consumo anual", col.main="blue", font.main=4)
  }
  if (numero_meses() < meses) {
    if (numero_meses() == 1) {
      p <-
      switch((mes[1]) + 1, "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Ju
      lio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre")
    }
    if (numero_meses() == 2) {
      c <-
      switch((mes[1]) + 1, "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Ju
      lio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre")
      d <-
      switch((mes[2]) + 1, "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Ju
      lio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre")
      p <- c(paste(c), paste(d))
    }
    if (numero_meses() == 3) {
      c <-
      switch((mes[1]) + 1, "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Ju
      lio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre")
    }
  }
}

```

```

        d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        e<-
switch((mes[3])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        p<-c(paste(c),paste(d),paste(e))
    }

    if (numero_meses()==4){

        d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        e<-
switch((mes[4])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        p<-c(" ",paste(d)," ",paste(e))
    }

    if (numero_meses()==5){

        d<-
switch((mes[1])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        e<-
switch((mes[3])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        f<-
switch((mes[5])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        p<-c(paste(d)," ",paste(e)," ",paste(f))
    }

    if (numero_meses()==6){

        d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        e<-
switch((mes[4])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        f<-
switch((mes[6])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")
    }

```

```

        p<-c(" ",paste(d)," ",paste(e)," ",paste(f)," ")
    }

    if (numero_meses()==7){

        d<-
switch((mes[1])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        e<-
switch((mes[3])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        f<-
switch((mes[5])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        g<-
switch((mes[7])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        p<-c(paste(d)," ",paste(e)," ",paste(f)," ",paste(g))
    }

    if (numero_meses()==8){

        d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        e<-
switch((mes[4])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        f<-
switch((mes[6])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        g<-
switch((mes[8])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        p<-c(" ",paste(d)," ",paste(e)," ",paste(f)," ",paste(g))
    }

    if (numero_meses()==9){

        d<-
switch((mes[1])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        e<-
switch((mes[3])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")
    }

```



```

        f<-
switch((mes[5])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        g<-
switch((mes[7])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        h<-
switch((mes[9])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        p<-c(paste(d)," ",paste(e)," ",paste(f)," ",paste(g),"
",paste(p))
    }

    if (numero_meses()==10){

        d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        e<-
switch((mes[4])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        f<-
switch((mes[6])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        g<-
switch((mes[8])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        h<-
switch((mes[10])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        p<-c(" ",paste(d)," ",paste(e)," ",paste(f)," ",paste(g),"
",paste(h))
    }

    if (numero_meses()==11){

        d<-
switch((mes[1])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        e<-
switch((mes[3])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        f<-
switch((mes[5])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")
    
```

```

        g<-
switch((mes[7])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        h<-
switch((mes[9])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        i<-
switch((mes[11])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        p<-c(paste(d)," ",paste(e)," ",paste(f)," ",paste(g),"
",paste(p)," ")

    }

    if (numero_meses()==12){

        d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        e<-
switch((mes[4])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        f<-
switch((mes[6])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        g<-
switch((mes[8])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        h<-
switch((mes[10])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        i<-
switch((mes[10])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        p<-c(" ",paste(d)," ",paste(e)," ",paste(f)," ",paste(g),"
",paste(h)," ",paste(i))

    }

    barplot(y,col="blue",width=1,xlab="meses",ylab="Energía consumida (kWh)",border="blue",ylim=c(0,limite),xlim=c(0,17),names.arg=p)

    title(main="Consumo anual", col.main="blue", font.main=4)

}

```

```
}

}))

output$aviso_3<-renderText({

  if (numero_meses()<=12){

    paste("No hay datos de los datos mensuales del año pasado, por
lo que no es posible realizar la comparación")

  }

}))

#####DATOS SIGNIFICATIVOS#####

#mes del año de mayor consumo

valor_mes_mayor_consumo<-reactive({

  aux<-datos_mensuales()[numero_meses(),2]

  meses<-as.numeric(as.character(aux))

  if (numero_meses()>meses){

    a<- datos_mensuales()[(numero_meses()-meses):(numero_meses()),3]

    y<-as.numeric(as.character(a))

    maximo<-max(y)/1000

  }

  else

    a<- datos_mensuales()[1:(numero_meses()),3]

  y<-as.numeric(as.character(a))

  maximo<-max(y)/1000

})

}))
```

```

output$consumo_mes_max<-renderText ({

  if (numero_meses ()>1) {

    aux<-valor_mes_mayor_consumo ()

    a<-as.numeric (as.character (aux))

    b<-round (a, 2)

    paste (b, "kWh")

  }

})

output$nombre_mes_max<-renderText ({

  if (numero_meses ()>1) {

    aux<-datos_mensuales () [numero_meses (), 2]

    meses<-as.numeric (as.character (aux))

    if (numero_meses ()>meses) {

      a<- datos_mensuales () [(numero_meses () -
meses) : (numero_meses ()) , 3]

      y<-as.numeric (as.character (a))

      fila_max_ultimo_año<-which.max (y)      #nos dara el mes de enero
como 1, y antes el mes de enero lo hemos definido como mes 0

      nombre<-
switch (fila_max_ultimo_año, "Enero", "Febrero", "Marzo", "Abril", "Mayo", "J
unio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre")

    }

    else

      a<- datos_mensuales () [1: (numero_meses ()) , 3]

      y<-as.numeric (as.character (a))

      fila_max_ultimo_año<-which.max (y)

      nombre<-
switch (fila_max_ultimo_año, "Enero", "Febrero", "Marzo", "Abril", "Mayo", "J
unio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre")

```

```

    }
  })

#GRAFICA DEL MES DE MAYOR CONSUMO

#cálculo del numero de días del año que han pasado, esto se hace con
el calculo de frecuencias acumulativas

#recordemos que calculamos con los ultimos doce meses, y nos los po-
ne ordenados desde enero hasta diciembre frecuencia_acumulativa<-
reactive({

tabla_numero_dias_ultimos_12_acumulativo<-reactive({
  aux<-datos_diarios() [numero_dias(),3]
  dias<-as.numeric(as.character(aux))
  if (numero_años()>1){
    zaux<-datos_diarios() [((numero_dias()-dias-
numero_dias_año_pasado()):numero_dias()-dias),2]
    vector_meses<-as.numeric(as.character(zaux))
    tabla_frecuencias<-table(zaux)
    tabla<-data.table(tabla_frecuencias)
    cumsum(tabla)
  }
  else
    if (numero_dias()>dias){
      zaux<-datos_diarios() [1:(numero_dias()-dias),2]
      vector_meses<-as.numeric(as.character(zaux))
      tabla_frecuencias<-table(zaux)
      tabla<-data.table(tabla_frecuencias)
      cumsum(tabla)
    }
  }
})

```

```
    }  
    else  
      z=0  
  
  })  
  
  tabla_numero_dias_cada_año<-reactive({  
    zaux<-datos_diarios() [ (1:numero_dias()),1]  
    vector_años<-as.numeric(as.character(zaux))  
    tabla_frecuencias<-table(vector_años)  
  })  
  
  tabla_numero_dias_año_acumulativo<-reactive({  
    aux<-datos_diarios() [1:numero_dias(),1]  
    dias<-as.numeric(as.character(aux))  
    tabla_frecuencias<-table(dias)  
    tabla<-data.table(tabla_frecuencias)  
    a<-cumsum(tabla)  
  })  
  
  output$grafica_mes_max<- renderPlot({  
  
    aux<-datos_diarios() [numero_dias(),3]  
    dias<-as.numeric(as.character(aux))  
    baux<-datos_diarios() [numero_dias(),2]  
    meses<-as.numeric(as.character(baux))
```

```

if (numero_meses()>1){
  if (numero_meses()>meses){
    a<- datos_mensuales()[(numero_meses()-
meses):(numero_meses()),3]
    y<-as.numeric(as.character(a))
    fila_max_ultimo_año<-which.max(y)
    p<-tabla_numero_dias_ultimos_12_meses()[fila_max_ultimo_año]
    dias_mes_max<-as.numeric(as.character(p))
    if (fila_max_ultimo_año==1){
      dias_hasta_mes_max=0
    }
    if (fila_max_ultimo_año>1){
      g<-
tabla_numero_dias_ultimos_12_acumulativo()[fila_max_ultimo_año-1]
      dias_hasta_mes_max<-as.numeric(as.character(g))
    }

    fila<-nrow(tabla_numero_dias_año_acumulativo())

    if (fila>1){
      w<-tabla_numero_dias_año_acumulativo()[fila-1]
      dias_hasta_año_actual<-as.numeric(as.character(w))
    }
    if (fila==0){
      dias_hasta_año_actual=0
    }

    b<-
da-
tos_diarios()[((dias_hasta_año_actual+dias_hasta_mes_max+1):(dias_hast
a_año_actual+dias_hasta_mes_max+dias_mes_max-1)),4]

    z<-as.numeric(as.character(b))/1000

    vector<-c(1:length(z))
  
```

```

    if (length(z)==28){

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ")

    }

    if (length(z)==29){

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ",vector[29])

    }

    if (length(z)==30){

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ",vector[29]," ")

    }

    if (length(z)==31){

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ",vector[29],"
",vector[31])

    }

    if (length(z)<28){

        b<-
da-
tos_diarios()[((dias_hasta_año_actual+dias_hasta_mes_max):(dias_hasta_
año_actual+dias_hasta_mes_max+dias_mes_max-1)),3]

        p<-as.numeric(as.character(b))

        var_x=c(p[1]:p[length(p)])

    }

    barplot(z,col="blue",width=1,xlab="dias",ylab="Energía con-
sumida
(kWh)",border="blue",ylim=c(0,max(z)+5),xlim=c(0,37),names.arg=var_x)

}

```



```

if (numero_meses() <= meses) {
  a <- datos_mensuales()[1:(numero_meses()-1), 3]
  y <- as.numeric(as.character(a))
  c <- c(1:meses) * 0
  d <- c(c, y)
  fila_max_ultimo_año <- which.min(d)
  e <- c(c, tabla_numero_dias_ultimos_12_meses())
  p <- d[fila_max_ultimo_año]
  dias_mes_max <- as.numeric(as.character(p))
  f <- c(c, tabla_numero_dias_ultimos_12_acumulativo())

  if (fila_max_ultimo_año == 1) {
    dias_hasta_mes_max = 0
  }

  if (fila_max_ultimo_año > 1) {
    g <- f[fila_max_ultimo_año - 1]
    dias_hasta_mes_max <- as.numeric(as.character(g))
  }

  b <-
da-
tos_diarios()[((dias_hasta_mes_max + 1):(dias_hasta_mes_max + dias_mes_max
- 1)), 4]

  z <- as.numeric(as.character(b)) / 1000
  vector <- c(1:length(z))

  if (length(z) < 28) {
    b <-
da-
tos_diarios()[((dias_hasta_año_actual + dias_hasta_mes_max + 1):(dias_hast
a_año_actual + dias_hasta_mes_max + dias_mes_max - 1)), 3]

    p <- as.numeric(as.character(b))

    var_x <- c(p[1]:p[length(p)])
  }

```

```

    }

    if (length(z)==28){

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ")

    }

    if (length(z)==29){

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ",vector[29])

    }

    if (length(z)==30){

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ",vector[29]," ")

    }

    if (length(z)==31){

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ",vector[29],"
",vector[31])

    }

    barplot(z,col="blue",width=1,xlab="dias",ylab="Energía consu-
mida
(kWh)",border="blue",ylim=c(0,max(z)+5),xlim=c(0,37),names.arg=var_x)

}

title(main="Mes de mayor consumo de energía",col.main="blue",
font.main=4)

}

}))

```

```
#calculo de factura de ese mes

output$gasto_mes_max<-renderText({

  if (numero_meses())>1){

    a<- valor_mes_mayor_consumo()

    b<-a*precio_electricidad

    consumo<-round(b,2)

    aux<-datos_diarios()[numero_dias(),3]

    dias<-as.numeric(as.character(aux))

    baux<-datos_diarios()[numero_dias(),2]

    meses<-as.numeric(as.character(baux))

    if (numero_meses())>meses){

      a<- datos_mensuales()[ (numero_meses()-
meses):(numero_meses()),3]

    }

    if (numero_meses())<=meses){

      a<- datos_mensuales()[1:(numero_meses()),3]

    }

    y<-as.numeric(as.character(a))

    fila_max_ultimo_año<-which.max(y)

    p<-tabla_numero_dias_ultimos_12_meses()[fila_max_ultimo_año]

    dias_mes_max<-as.numeric(as.character(p))

    pot<-potencia_contratada*dias_mes_max*precio_potencia

    potencia<-round(pot,2)

    imp<-(consumo+potencia)*impuesto_electricidad

    impuesto<-round(imp,2)

  }
```

```

total<-consumo+potencia+impuesto+serv_prot+alquiler_equipos

con_iva<-round(total*iva,2)

factura_total<-total+con_iva

paste(factura_total,"Euros")

}

})

output$aviso_max<-renderText({

  if (numero_meses()==1){

    a<-paste("Al tener datos sobre un único mes no tiene sentido ha-
blar del mes de máximo consumo")

  }

})

#mes del año de menor consumo (entre los ultimos doce meses)

valor_mes_minimo_consumo<-reactive({

  aux<-datos_mensuales()[numero_meses(),2]
  meses<-as.numeric(as.character(aux))
  if (numero_meses())>meses){
    a<- datos_mensuales()[ (numero_meses()-meses):(numero_meses()-
1),3]
    y<-as.numeric(as.character(a))
    minimo<-min(y)/1000
  }
  else
    a<- datos_mensuales()[1:(numero_meses()-1),3]
  y<-as.numeric(as.character(a))

```

```
minimo<-min(y)/1000

}))

output$consumo_mes_min<-renderText({

  if (numero_meses()>1){

    aux<-valor_mes_minimo_consumo()

    a<-as.numeric(as.character(aux))

    b<-round(a,2)

    paste(b,"kWh")

  }

}))

output$nombre_mes_min<-renderText({

  if (numero_meses()>1){

    aux<-datos_mensuales()[numero_meses(),2]

    meses<-as.numeric(as.character(aux))

    if (numero_meses()>meses){

      a<- datos_mensuales()[(numero_meses()-meses):(numero_meses()-
1),3]

      y<-as.numeric(as.character(a))

      fila_min_ultimo_año<-which.min(y) #nos dara el mes de enero
como 1, y antes el mes de enero lo hemos definido como mes 0

      nombre<-
switch(fila_min_ultimo_año,"Enero","Febrero","Marzo","Abril","Mayo","J
unio","Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")
```

```

    }

    else

        a<- datos_mensuales()[1:(numero_meses()),3]

        y<-as.numeric(as.character(a))

        fila_min_ultimo_año<-which.min(y)

        nombre<-
switch(fila_min_ultimo_año,"Enero","Febrero","Marzo","Abril","Mayo","J
unio","Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    }

}))

```

#grafica del mes de menor consumo

```

output$grafica_mes_min<- renderPlot({

    aux<-datos_diarios()[numero_dias(),3]
    dias<-as.numeric(as.character(aux))
    baux<-datos_diarios()[numero_dias(),2]
    meses<-as.numeric(as.character(baux))
    if (numero_meses())>1){
        if (numero_meses())>meses){
            a<- datos_mensuales()[ (numero_meses()-meses):(numero_meses()-
1),3]
            y<-as.numeric(as.character(a))
            fila_min_ultimo_año<-which.min(y)
            p<-tabla_numero_dias_ultimos_12_meses()[fila_min_ultimo_año]
            dias_mes_min<-as.numeric(as.character(p))
            if (fila_min_ultimo_año==1){
                dias_hasta_mes_min=0
            }
            if (fila_min_ultimo_año>1){

```

```

g<-
tabla_numero_dias_ultimos_12_acumulativo()[fila_min_ultimo_año-1]

    dias_hasta_mes_min<-as.numeric(as.character(g))
}

fila<-nrow(tabla_numero_dias_año_acumulativo())

if (fila>1){
    w<-tabla_numero_dias_año_acumulativo()[fila-1]
    dias_hasta_año_actual<-as.numeric(as.character(w))
}

if (fila==0){
    dias_hasta_año_actual=0
}

b<-
da-
tos_diarios()[((dias_hasta_año_actual+dias_hasta_mes_min):(dias_hasta_
año_actual+dias_hasta_mes_min+dias_mes_min-1)),4]

z<-as.numeric(as.character(b))/1000

vector<-c(1:length(z))

if (length(z)==28){
    var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ",vector[25]," ",vector[27]," ")
}

if (length(z)==29){
    var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ",vector[25]," ",vector[27]," ",vector[29])
}

if (length(z)==30){

```

```

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ",vector[29]," ")
    }

    if (length(z)==31){

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ",vector[29],"
",vector[31])

    }

    if (length(z)<28){

        b<-
da-
tos_diarios()(((dias_hasta_año_actual+dias_hasta_mes_min):(dias_hasta_
año_actual+dias_hasta_mes_min+dias_mes_min-1)),3]

        p<-as.numeric(as.character(b))

        var_x=c(p[1]:p[length(p)])

    }

    barplot(z,col="blue",width=1,xlab="dias",ylab="Energía con-
sumida
(kWh)",border="blue",ylim=c(0,max(z)+5),xlim=c(0,37),names.arg=var_x)

}

if (numero_meses()<=meses){

    a<- datos_mensuales()[1:(numero_meses()-1),3]

    y<-as.numeric(as.character(a))

    c<-c(1:meses)*0

    d<-c(c,y)

    fila_min_ultimo_año<-which.min(d)

    e<-c(c,tabla_numero_dias_ultimos_12_meses())

    p<-d[fila_min_ultimo_año]

    dias_mes_min<-as.numeric(as.character(p))

    f<-c(c,tabla_numero_dias_ultimos_12_acumulativo())

```



```

    if (fila_min_ultimo_año==1){
        dias_hasta_mes_min=0
    }
    if (fila_min_ultimo_año>1){
        g<-f[fila_min_ultimo_año-1]
        dias_hasta_mes_min<-as.numeric(as.character(g))
    }

    b<-
da-
tos_diarios()[((dias_hasta_mes_min):(dias_hasta_mes_min+dias_mes_min-
1)),4]

    z<-as.numeric(as.character(b))/1000
    vector<-c(1:length(z))
    if (length(z)<28){
        b<-
da-
tos_diarios()[((dias_hasta_año_actual+dias_hasta_mes_min):(dias_hasta_
año_actual+dias_hasta_mes_min+dias_mes_min-1)),3]

        p<-as.numeric(as.character(b))
        var_x=c(p[1]:p[length(p)])
    }
    if (length(z)==28){
        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ")
    }
    if (length(z)==29){
        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ",vector[29])
    }
    if (length(z)==30){

```

```

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ",vector[29]," ")
    }

    if (length(z)==31){

        var_x=c(vector[1]," ",vector[3]," ",vector[5],"
",vector[7]," ",vector[9]," ",vector[11]," ",vector[13],"
",vector[15]," ",vector[17]," ",vector[19]," ",vector[21],"
",vector[23]," ", vector[25]," ",vector[27]," ",vector[29],"
",vector[31])

    }

    barplot(z,col="blue",width=1,xlab="dias",ylab="Energía consu-
mida
(kWh)",border="blue",ylim=c(0,max(z)+5),xlim=c(0,37),names.arg=var_x)

}

    title(main="Mes de menor consumo de energía",col.main="blue",
font.main=4)

}

}))

#cálculo de factura

output$gasto_mes_min<-renderText({

    if (numero_meses())>1) {

        a<- valor_mes_minimo_consumo()

        cons<-a*precio_electricidad

        consumo<-round(cons,2)

        aux<-datos_diarios()[numero_dias(),3]

        dias<-as.numeric(as.character(aux))

        baux<-datos_diarios()[numero_dias(),2]

        meses<-as.numeric(as.character(baux))
    }
}

```

```

a<- datos_mensuales()[(numero_meses()-meses):(numero_meses()-
1),3]

y<-as.numeric(as.character(a))

fila_min_ultimo_año<-which.min(y)

p<-tabla_numero_dias_ultimos_12_meses()[fila_min_ultimo_año]

dias_mes_min<-as.numeric(as.character(p))

pot<-potencia_contratada*dias_mes_min*precio_potencia

potencia<-round(pot,2)

imp<-(consumo+potencia)*impuesto_electricidad

impuesto<-round(imp,2)

total<-consumo+potencia+impuesto+serv_prot+alquiler Equipos

con_iva<-round(total*iva,2)

factura_total<-total+con_iva

paste(factura_total,"Euros")

}

})

output$aviso_min<-renderText({

  if (numero_meses()==1){

    a<-paste("Al tener datos sobre un único mes no tiene sentido ha-
blar del mes de mínimo consumo")

  }

})

#Consumo maximo diario del año

valor_dia_max_consumo<-reactive({

```

```

if(numero_dias())>1){
  aux<-datos_diarios()[numero_dias(),3]
  dias_totales<-as.numeric(as.character(aux))
  filas<-nrow(tabla_numero_dias_cada_año())
  baux<-tabla_numero_dias_cada_año()[filas]
  dias_año<-as.numeric(as.character(baux))
  a<- datos_diarios()[ (numero_dias()-dias_año):(numero_dias()),4]
  y<-as.numeric(as.character(a))
  z<-max(y)/1000
}
})

output$valor_dia_max_consumo<-renderText({
  if (numero_dias())>1){

    a<-round(valor_dia_max_consumo(),2)
    paste(a,"kWh")

  }
})

#Día en el que se da el máximo consumo

output$dia_max<-renderText({
  if (numero_dias())>1){
    aux<-datos_diarios()[numero_dias(),3]
    dias_totales<-as.numeric(as.character(aux))
    filas<-nrow(tabla_numero_dias_cada_año())
    baux<-tabla_numero_dias_cada_año()[filas]
    dias_año<-as.numeric(as.character(baux))
    a<- datos_diarios()[ (numero_dias()-dias_año):(numero_dias()),4]

```

```

y<-as.numeric(as.character(a))

fila_dia_max<-which.max(y) #nos dara el mes de enero como 1, y
antes el mes de enero lo hemos definido como mes 0

longitud_vector<-length(y)

resta<-(longitud_vector)-(fila_dia_max)

año<-datos_diarios()[numero_dias()-resta,1]

mes<-datos_diarios()[numero_dias()-resta,2]

ai<-as.numeric(as.character(mes))

nombre_mes<-
switch(ai+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","
Agosto","Septiembre","Octubre","Noviembre","Diciembre")

dia<-datos_diarios()[numero_dias()-resta,3]

texto<-paste(dia,"de",nombre_mes,"del",año)

}

})

```

```
#grafica del dia de máximo consumo
```

```

output$grafica_dia_max<-renderPlot({
  if (numero_dias())>1){
    aux<-datos_diarios()[numero_dias(),3]
    dias_totales<-as.numeric(as.character(aux))
    filas<-nrow(tabla_numero_dias_cada_año())
    baux<-tabla_numero_dias_cada_año()[filas]
    dias_año<-as.numeric(as.character(baux))
    a<- datos_diarios()[(numero_dias()-dias_año):(numero_dias()),4]
    y<-as.numeric(as.character(a))

    fila_dia_max<-which.max(y) #nos dara el mes de enero como 1, y
antes el mes de enero lo hemos definido como mes 0

```

```

longitud_vector<-length(y)

resta<-(longitud_vector)-(fila_dia_max)

zaux<-datos_horarios()[numero_horas(),4]

horas<-as.numeric(as.character(zaux))

posicion_dias<-datos_horarios()[((numero_horas()-horas-
(resta*24)):(numero_horas()-horas-(resta*24)+23)),5]

z<-as.numeric(as.character(posicion_dias))/1000

barplot(z,col="blue",width=1,xlab="horas",ylab="Energía consumi-
da
(kWh)",border="blue",ylim=c(0,max(z)+0.5),xlim=c(0,28),names.arg=c("0h
"," ","2h"," ","4h"," ","6h"," ","8h"," ","10h"," ","12h"," ","14h","
","16h"," ","18h"," ","20h"," ","22h"," "))

title(main="Día de mayor consumo",col.main="blue", font.main=4)

}

})

```

```
#Consumo minimo diario del año
```

```

valor_dia_menor_consumo<-reactive({

  if (numero_dias())>1){

    aux<-datos_diarios()[numero_dias(),3]

    dias_totales<-as.numeric(as.character(aux))

    filas<-nrow(tabla_numero_dias_cada_año())

    baux<-tabla_numero_dias_cada_año()[filas]

    dias_año<-as.numeric(as.character(baux))

    zaux<-datos_horarios()[1,4]

    horas_dia_1<-as.numeric(as.character(zaux))

    if (horas_dia_1==0){

```

```

        a<- datos_diarios()[(numero_dias()-dias_año):(numero_dias()-
1),4]

        y<-as.numeric(as.character(a))

    }

    if (horas_dia_1>0){

        if (numero_años(>1){

            a<- datos_diarios()[(numero_dias()-dias_año):(numero_dias()-
1),4]

            y<-as.numeric(as.character(a))

        }

        if (numero_años()==1){

            if (numero_dias(>2){

                a<- datos_diarios()[2:(numero_dias()-1),4]

                y<-as.numeric(as.character(a))

            }

        }

    }

    minimo<-min(y)/1000

}

}))

output$valor_dia_menor_consumo<-renderText({

    if (numero_dias(>1){

        a<-round(valor_dia_menor_consumo(),2)

        paste(a,"kWh")

    }

})

```

```

}))

#Día en el que se da el minimo consumo (quitando el dia actual)

output$dia_min<-renderText({
  if (numero_dias())>1){
    aux<-datos_diarios()[numero_dias(),3]
    dias_totales<-as.numeric(as.character(aux))
    filas<-nrow(tabla_numero_dias_cada_año())
    baux<-tabla_numero_dias_cada_año()[filas]
    dias_año<-as.numeric(as.character(baux))
    zaux<-datos_horarios()[1,4]
    horas_dia_1<-as.numeric(as.character(zaux))

    if (horas_dia_1==0){
      a<- datos_diarios()[ (numero_dias()-dias_año):(numero_dias()-
1),4]
      y<-as.numeric(as.character(a))
    }

    if (horas_dia_1>0){
      if (numero_años())>1){
        a<- datos_diarios()[ (numero_dias()-dias_año):(numero_dias()-
1),4]
        y<-as.numeric(as.character(a))
      }

      if (numero_años()==1){
        if (numero_dias())>2){
          a<- datos_diarios()[2:(numero_dias()-1),4]
          y<-as.numeric(as.character(a))
        }
      }
    }
  }
})

```



```

    }

    }

}

    fila_dia_min<-which.min(y) #nos dara el mes de enero como 1, y
antes el mes de enero lo hemos definido como mes 0

    longitud_vector<-length(y)

    resta<-(longitud_vector)-(fila_dia_min)

    año<-datos_diarios()[numero_dias()-1-resta,1]

    mes<-datos_diarios()[numero_dias()-1-resta,2]

    ai<-as.numeric(as.character(mes))

    nombre_mes<-
switch(ai+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","
Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    dia<-datos_diarios()[numero_dias()-1-resta,3]

    texto<-paste(dia,"de",nombre_mes,"del",año)

}

}))

#grafica del dia de minimo consumo

output$grafica_dia_min<-renderPlot({

    if (numero_dias())>1){

        aux<-datos_diarios()[numero_dias(),3]

        dias_totales<-as.numeric(as.character(aux))

        filas<-nrow(tabla_numero_dias_cada_año())

        baux<-tabla_numero_dias_cada_año()[filas]

        dias_año<-as.numeric(as.character(baux))

        zaux<-datos_horarios()[1,4]

        horas_dia_1<-as.numeric(as.character(zaux))
    }
}

```

```

    if (horas_dia_1==0){

      a<- datos_diarios()[(numero_dias()-dias_año):(numero_dias()-
1),4]

      y<-as.numeric(as.character(a))

    }

    if (horas_dia_1>0){

      if (numero_años(>1){

        a<- datos_diarios()[(numero_dias()-dias_año):(numero_dias()-
1),4]

        y<-as.numeric(as.character(a))

      }

      if (numero_años()==1){

        if (numero_dias(>2){

          a<- datos_diarios()[2:(numero_dias()-1),4]

          y<-as.numeric(as.character(a))

        }

      }

    }

    fila_dia_min<-which.min(y) #nos dara el mes de enero como 1, y
antes el mes de enero lo hemos definido como mes 0

    longitud_vector<-length(y)

    zaux<-datos_horarios()[numero_horas(),4]

    horas<-as.numeric(as.character(zaux))

    resta<-(longitud_vector)-(fila_dia_min)

    posicion_dias<-datos_horarios()[((numero_horas()-horas-
(resta*24)-24):(numero_horas()-horas-(resta*24)-1)),5]

    z<-(as.numeric(as.character(posicion_dias)))/1000

    barplot(z,col="blue",width=1,xlab="horas",ylab="Energía consumi-
da

```

```
(kWh)",border="blue",ylim=c(0,max(z)+1),xlim=c(0,28),names.arg=c("0h",
" ", "2h", " ", "4h", " ", "6h", " ", "8h", " ", "10h", " ", "12h", " ", "14h", "
", "16h", " ", "18h", " ", "20h", " ", "22h", " "))

    title(main="Día de menor consumo",col.main="blue", font.main=4)

  }

})
```

#####FACTURA DE ESTE MES#####

#factura del mes actual

```
cons1<-reactive({
  aux<-datos_mensuales()[numero_meses(),3]
  b<-as.numeric(as.character(aux))/1000
  round(b,2)
})
```

```
output$cons1<-renderText({
  paste("·Energía consumida: ",cons1(),"kWh")
})
```

```
precio_cons1<-reactive({
  b<-cons1()*precio_electricidad
  c<-round(b,2)
})
```

```
output$precio_cons1<-renderText({
  consumo<-cons1()
```

```
precio<-precio_cons1()

texto<-paste(consumo,"kWh x",precio_electricidad,"Euros/kWh
=",precio,"Euros")

})

numero_dias_mes1<-reactive({
  aux<-datos_diarios()[numero_dias(),3]
  b<-as.numeric(as.character(aux))
})

precio_potencial<-reactive({
  aux<-potencia_contratada*precio_potencia*numero_dias_mes1()
  b<-as.numeric(as.character(aux))
  c<-round(b,2)
})

output$precio_potencial<-renderText({
  aux<-precio_potencial()
  dias<-numero_dias_mes1()
  texto<-paste(potencia_contratada,"kW x",dias,"dias
x",precio_potencia,"Euro/kW y dia =",aux, "Euros")
})

impto_electricidad1<-reactive({
  aux<- (precio_cons1()+precio_potencial())*impuesto_electricidad
  v<-round(aux,2)
})

output$impto_electricidad1<-renderText({
```

```
impuesto<-impto_electricidad1()

cons1<-precio_cons1()

precio1<-precio_potencial()

texto<-paste(" 4.864% x (",cons1,"+",precio1,")","Euros x 1.05113
=",impuesto,"Euros")

})

servicio_proteccion1<-reactive({

  pago=serv_prot

})

output$servicio_proteccion1<-renderText({

  a<-servicio_proteccion1()

  b<-paste(a, "Euros/mes")

})

output$alquiler_equipos1<-reactive({

  a=alquiler_equipos

  paste(a,"Euros")

})

total1<-reactive({

  a<-
pre-
cio_potencial()+precio_cons1()+impto_electricidad1()+servicio_protecci
on1()+alquiler_equipos

  b<-round(a,2)

})

output$total1<-renderText({

  paste(total1(),"Euros")

})
```

```
precio_ival<-reactive({
  a<- total1()*0.21
  b<-round(a,2)
})

output$precio_ival<-renderText({
  a<-precio_ival()
  paste(a,"Euros")
})

output$factura_total1<-renderText({
  total<-total1()+precio_ival()
  paste(total,"Euros")
})

output$nombre_mes1<-renderText({
  aux<-datos_mensuales()[numero_meses(),2]
  mes<-as.numeric(as.character(aux))
  nombre<-
switch(mes+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")
  paste("·Mes: ",nombre)
})

output$año2<-renderText({
  año<-datos_mensuales()[numero_meses(),1]
  paste("·Año: ",año)
})

#grafica del consumo de los ultimos 12 meses
```

```

output$grafica_consumo_12meses<-renderPlot({
  if (numero_meses()>12){
    a<-datos_mensuales()[ (numero_meses()-11):(numero_meses()),3]
    y<-as.numeric(as.character(a))/1000
    b<-datos_mensuales()[ (numero_meses()-11):(numero_meses()),2]
    mes<-as.numeric(as.character(b))
    d<-
    switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
    lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")
    e<-
    switch((mes[4])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
    lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")
    f<-
    switch((mes[6])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
    lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")
    g<-
    switch((mes[8])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Ju
    lio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")
    h<-
    switch((mes[10])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","J
    ulio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")
    i<-
    switch((mes[12])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","J
    ulio","Agosto","Septiembre","Octubre","Noviembre","Diciembre")
    p<-c(" ",paste(d)," ",paste(e)," ",paste(f)," ",paste(g),"
    ",paste(h)," ",paste(i))
    maximo<-max(y)
    limite<-maximo+50
    barplot(y,col="blue",xlab="meses",ylab="Energía consumida
    (kWh)",border="blue",ylim=c(0,limite),xlim=c(0,15),names.arg=p)
    title(main="Historial de consumo de los últimos 12 meses",
    col.main="blue", font.main=4)
  }
  if (numero_meses()<=12){
    a<-datos_mensuales()[1:(numero_meses()),3]
    y<-as.numeric(as.character(a))/1000
    b<-datos_mensuales()[1:(numero_meses()),2]
  }
}

```

```

mes<-as.numeric(as.character(b))

if (numero_meses()==1){

    p<-
switch((mes[1])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

}

if (numero_meses()==2){

    c<-
switch((mes[1])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    p<-c(paste(c),paste(d))

}

if (numero_meses()==3){

    c<-
switch((mes[1])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    e<-
switch((mes[3])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    p<-c(paste(c),paste(d),paste(e))

}

if (numero_meses()==4){

    d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    e<-
switch((mes[4])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    p<-c(" ",paste(d)," ",paste(e))

}

if (numero_meses()==5){

```



```

    d<-
switch((mes[1])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    e<-
switch((mes[3])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    f<-
switch((mes[5])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    p<-c(paste(d)," ",paste(e)," ",paste(f))
}

if (numero_meses()==6){

    d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    e<-
switch((mes[4])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    f<-
switch((mes[6])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    p<-c(" ",paste(d)," ",paste(e)," ",paste(f)," ")
}

if (numero_meses()==7){

    d<-
switch((mes[1])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    e<-
switch((mes[3])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    f<-
switch((mes[5])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    g<-
switch((mes[7])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    p<-c(paste(d)," ",paste(e)," ",paste(f)," ",paste(g))
}

if (numero_meses()==8){

```

```

    d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    e<-
switch((mes[4])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    f<-
switch((mes[6])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    g<-
switch((mes[8])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    p<-c(" ",paste(d)," ",paste(e)," ",paste(f)," ",paste(g))

}

if (numero_meses()==9){

    d<-
switch((mes[1])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    e<-
switch((mes[3])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    f<-
switch((mes[5])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    g<-
switch((mes[7])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    h<-
switch((mes[9])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    p<-c(paste(d)," ",paste(e)," ",paste(f)," ",paste(g),"
",paste(p))

}

if (numero_meses()==10){

    d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

    e<-
switch((mes[4])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

```

```

f<-
switch((mes[6])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

g<-
switch((mes[8])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

h<-
switch((mes[10])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

p<-c(" ",paste(d)," ",paste(e)," ",paste(f)," ",paste(g),"
",paste(h))

}

if (numero_meses()==11){

d<-
switch((mes[1])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

e<-
switch((mes[3])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

f<-
switch((mes[5])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

g<-
switch((mes[7])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

h<-
switch((mes[9])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

i<-
switch((mes[11])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

p<-c(paste(d)," ",paste(e)," ",paste(f)," ",paste(g),"
",paste(p)," ")

}

if (numero_meses()==12){

d<-
switch((mes[2])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

e<-
switch((mes[4])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

```

```

        f<-
switch((mes[6])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        g<-
switch((mes[8])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        h<-
switch((mes[10])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        i<-
switch((mes[10])+1,"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",
"Agosto","Septiembre","Octubre","Noviembre","Diciembre")

        p<-c(" ",paste(d)," ",paste(e)," ",paste(f)," ",paste(g),"
",paste(h)," ",paste(i))

    }

    maximo<-max(y)

    limite<-maximo+50

    barplot(y,col="blue",xlab="meses",ylab="Energía consumida
(kWh)",border="blue",ylim=c(0,limite),xlim=c(0,15),names.arg=p)

    title(main="Historial de consumo de los últimos 12 meses",
col.main="blue", font.main=4)

    }

    })

#factura del mes anterior

cons2<-reactive({
    if (numero_meses())>1{
        aux<-datos_mensuales()[(numero_meses()-1),3]

        b<-as.numeric(as.character(aux))/1000

        round(b,2)
    }
})

```

```

    }
  })

  output$cons2<-renderText ({
    if (numero_meses ()>1) {
      paste ("·Energía consumida: ", cons2 (), "kWh")
    }
  })

  precio_cons2<-reactive ({
    if (numero_meses ()>1) {
      b<-cons2 ()*precio_electricidad
      c<-round (b, 2)
    }
  })

  output$precio_cons2<-renderText ({
    if (numero_meses ()>1) {
      consumo<-cons2 ()
      precio<-precio_cons2 ()
      texto<-paste (consumo, "kWh x", precio_electricidad, "Euros/kWh
      =", precio, "Euros")
    }
  })

  numero_dias_mes2<-reactive ({
    if (numero_meses ()>1) {
      aux<-datos_diarios () [numero_dias (), 3]
    }
  })

```

```

        b<-as.numeric(as.character(aux))

        c<-datos_diarios()[ (numero_dias()-b),3]

        d<-as.numeric(as.character(c))

    }

})

precio_potencia2<-reactive({

    if (numero_meses())>1){

        aux<-potencia_contratada*precio_potencia*numero_dias_mes2()

        b<-as.numeric(as.character(aux))

        c<-round(b,2)

    }

})

output$precio_potencia2<-renderText({

    if (numero_meses())>1){

        aux<-precio_potencia2()

        dias<-numero_dias_mes2()

        texto<-paste(potencia_contratada,"kW x",dias,"dias
x",precio_potencia,"Euro/kW y dia =",aux, "Euros")

    }

})

impto_electricidad2<-reactive({

    if (numero_meses())>1){

        aux<- (precio_cons2()+precio_potencia2())*impuesto_electricidad

        v<-round(aux,2)

    }

})

```

```
output$impto_electricidad2<-renderText({
  if (numero_meses()>1){
    impuesto<-impto_electricidad2()
    cons2<-precio_cons2()
    precio2<-precio_potencia2()
    texto<-paste("4.864% x (",cons2,"+",precio2,")", "Euros x
1.05113 =",impuesto,"Euros")
  }
})

servicio_proteccion2<-reactive({

  pago=serv_prot

})

output$servicio_proteccion2<-renderText({

  a<-servicio_proteccion2()
  b<-paste(a, "Euros/mes")

})

output$alquiler_equipos2<-reactive({

  a=alquiler_equipos

  b<-paste(a, "Euros")

})

total2<-reactive({

  if (numero_meses()>1){
```

```
      a<-
pre-
cio_potencia2()+precio_cons2()+impto_electricidad2()+servicio_protecci
on2()+alquiler_equipos

      round(a,2)

    }

  })

output$total2<-renderText({

  if (numero_meses())>1){

    a<-total2()

    paste(a,"Euros")

  }

})

precio_iva2<-reactive({

  if (numero_meses())>1){

    a<-total2()*0.21

    b<-round(a,2)

  }

})

output$precio_iva2<-renderText({

  if (numero_meses())>1){

    a<-precio_iva2()

    paste(a,"Euros")

  }

})

output$factura_total2<-renderText({

  if (numero_meses())>1){

    total<-total2()+precio_iva2()
```



```
      paste(total, "Euros")
    }
  })

output$nombre_mes2<-renderText({
  if (numero_meses()>1){
    aux<-datos_mensuales()[numero_meses()-1,2]
    mes<-as.numeric(as.character(aux))

    nombre<-
switch(mes+1, "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio", "Julio",
"Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre")

    paste("·Mes: ", nombre)
  }
})

output$año3<-renderText({
  if (numero_meses()>1){
    año<-datos_mensuales()[numero_meses()-1,1]
    paste("·Año: ", año)
  }
})

output$aviso_factura<-renderText({
  if (numero_meses()==1){
    a<-paste("No hay datos del mes anterior")
  }
})

})
```

