

Trabajo obligatorio: Tarifas y consumo eléctricos

Actualizado el 15 de enero de 2022

7.1. Objetivo del trabajo

Ante el gran aumento del precio de la electricidad en el mercado mayorista en los últimos meses, el cambio en la metodología para el cálculo de los peajes de transporte y distribución de electricidad que entró en vigor el 1 de junio de este año, y el gran desconocimiento general del funcionamiento del mercado eléctrico en España¹, una pequeña empresa dedicada a la realización de instalaciones eléctricas domésticas, Ineldo S. L., quiere ofrecer también a sus clientes servicios de asesoría sobre sus hábitos de consumo y de contratación de tarifas eléctricas.

Actualmente, un consumidor doméstico recibe su suministro eléctrico a través de una empresa *distribuidora de electricidad*, que no puede ser elegida por el consumidor (en Aragón, por ejemplo, esa empresa es Edistribución Redes Digitales², filial de Endesa³). Por ello, no es la empresa distribuidora la que factura al cliente, sino una empresa *comercializadora de electricidad*. Existen dos tipos de comercializadoras de electricidad: las del *mercado regulado* y las del *mercado libre*.

Las comercializadoras del mercado regulado, también conocidas como *comercializadoras de referencia*, ofrecen a sus abonados únicamente una tarifa, denominada *precio voluntario al pequeño consumidor* (PVPC), cuyos términos están fijados por el Gobierno de España y están basados en costes fijos (basados principalmente en la potencia contratada por el consumidor) y en los costes variables (basados en el consumo eléctrico real de ese consumidor). Estos costes de consumo se calculan a través de un contador que registra, cada hora de cada día, el consumo eléctrico realizado en esa hora. En este trabajo, vamos a medir el consumo de energía eléctrica en kWh. A estos consumos horarios se les aplica un precio de la energía eléctrica establecido el día anterior en una subasta entre las compañías productoras de electricidad, en la que se establece, para cada hora del día posterior, un precio distinto de la energía eléctrica, en función de la oferta y demanda previstas. Esto conlleva que el precio de la electricidad varíe de una hora para otra en un mismo día, y también entre las mismas horas de días distintos. En Aragón, la comercializadora de referencia que aplica el PVPC es Energía XXI⁴, también del grupo Endesa.

Por último, las comercializadoras del mercado libre tienen total libertad para fijar los precios que cobran a sus abonados. Existen tarifas en las que el precio de la electricidad en €/kWh es constante durante un periodo de tiempo establecido en el contrato (normalmente, un año), otras en las que los precios se reparten en dos o tres franjas horarias y son constantes dentro de esa franja, otras en las que

¹<https://www.cnmc.es/file/306014/download>

²<https://www.edistribucion.com/>

³<https://www.endesa.com/>

⁴<https://www.energiaxxi.com/>

ofrece consumo gratuito durante una o dos horas diarias, utilizándose otro tipo de precios en las horas restantes, etc. La Comisión Nacional de los Mercados y la Competencia (CNMC) lista actualmente un total 72 ofertas de un total de 39 comercializadoras distintas⁵.

El proyecto a desarrollar como parte de este trabajo pretende ser una primera versión de un programa C++ destinado a analizar los consumos eléctricos de los clientes de la empresa de instalaciones eléctricas Ineldo S. L. así como los importes de las facturas eléctricas (**considerando únicamente la parte del consumo**) aplicando la tarifa eléctrica regulada (PVPC) y las tarifas ofrecidas por algunas compañías comercializadoras del mercado libre.

Para ello, se va a trabajar con dos tipos de ficheros: los que recogen la información sobre el precio que ha tenido la electricidad en la tarifa PVPC cada hora de cada día y los que contienen consumos eléctricos realizados por los clientes a los que Ineldo S. L. quiere asesorar. En la sección 7.2 se explica con más detalle de qué ficheros se trata y cuál es su contenido.

El programa a diseñar **debe permitir al usuario del mismo especificar de forma interactiva el nombre de un cliente y un periodo de análisis, expresado como meses entre enero y noviembre de este año, y generar un informe** con estadísticas sobre la tarifa PVPC en general, el importe económico del consumo del cliente en el periodo seleccionado (suponiendo que está acogido al PVPC) y el importe que habría tenido el mismo consumo de haber contratado otras ofertas en el mercado libre. El contenido concreto del informe se detalla en la sección 7.3.

El programa diseñado estará **integrado por varios módulos, con distintas funciones especializadas** en la resolución de problemas de pequeña entidad. Cada una de ellas estará convenientemente especificada; la lectura de estas especificaciones debe ser suficiente para comprender su comportamiento sin necesidad de consultar el código. El código de cada función debe constar de unas pocas líneas. El programa deberá ser fácilmente legible y estará adecuadamente documentado y presentado. **El diseño de este programa exige poner en práctica una buena parte de los conocimientos aprendidos a lo largo de toda la asignatura.**

En el repositorio de GitHub <https://github.com/progl-eina/trabajo-2021-22> se encuentra el material de partida para la realización de este trabajo (estructura de directorios, fichero «Makefile» y ficheros de datos con los que trabajar).

Antes de comenzar a realizar este trabajo, se recomienda la lectura completa del enunciado, con objeto de obtener una idea del conjunto de las tareas que la componen y de la carga de trabajo que puede suponer.

7.2. Ficheros de datos eléctricos

7.2.1. Fichero de precios horarios regulados PVPC

El fichero «datos/tarifas-2021-ene-nov.csv» ha sido descargado del Sistema de Información del Operador del Sistema de Red Eléctrica de España⁶. Recoge la información sobre el precio que ha tenido la electricidad en la tarifa PVPC cada hora de cada día desde el 1 de enero hasta el 30 de noviembre de 2021. La sintaxis que sigue y el contenido son los ofrecidos por Red Eléctrica de España, con la salvedad de que se han modificado los datos correspondientes a los días 28 de marzo (día de inicio del horario de verano y que tuvo solo 23 horas) y 31 de octubre (día en que acabó el horario de verano y que tuvo 25 horas), para que contengan exactamente 24 datos horarios, como el resto de los días.

⁵<https://comparador.cnmc.gob.es/>

⁶https://www.esios.ree.es/es/analisis/1001?vis=1&start_date=01-01-2021T00:00&end_date=30-11-2021T23:00&compare_start_date=31-12-2020T00:00&groupby=hour&compare_indicators=1013&geoids=8741

La sintaxis que sigue este fichero se muestra a continuación utilizando notación Backus-Naur:

```

<fichero-precios> ::= <cabecera> {<datos-precios-horarios>}
<cabecera> ::= "id;name;geoid;geoname;value;datetime" <fin-línea>
<datos-precios-horarios> ::= <id> <sep> <name> <sep> <geoid> <sep> <geoname> <sep>
                           <precio> <sep> <datetime> <fin-línea>
<id> ::= "1001" | "1013"
<name> ::= "Término de facturación de energía activa del PVPC peaje por defecto"
          | "Término de facturación de energía activa del PVPC 2.0TD"
<geoid> ::= "" | "8741"
<geoname> ::= "" | "Península"
<precio> ::= literal-real
<datetime> ::= <año> "-" <mes> "-" <día> "T" <hora> ":" <minutos> <dif-utc>
<año> ::= literal-entero
<mes> ::= literal-entero
<día> ::= literal-entero
<hora> ::= literal-entero
<minutos> ::= "00"
<dif-utc> ::= "+01:00" | "+02:00"
<fin-línea> ::= "\n"
<sep> ::= ";"

```

El fichero comienza con una cabecera que ocupa una línea y le siguen líneas con información sobre los precios horarios de cada una de las horas de cada día desde el 1 de enero hasta el 20 de noviembre de 2021. Cada línea, además de los datos relevantes para este trabajo (<precio>, <año>, <mes>, <día> y <hora>, que se muestran en negrita en el esquema sintáctico) contiene otro tipo de información en la que no estaremos interesados (como identificador y nombre del indicador eléctrico seleccionados para la descarga del fichero, identificador y nombre del ámbito geográfico de aplicación del mismo, minutos o diferencia entre la fecha y hora utilizadas y el tiempo universal coordinado). Los datos de cada línea del fichero están separados entre sí por el carácter punto y coma (';'). La fecha y la hora se han codificado como un único dato siguiendo la ISO 8601⁷, por lo que los valores individuales de <año>, <mes>, <día> y <hora> están separados entre sí por otros caracteres ('-' y 'T').

Los datos sobre el precio en cada hora están expresados en €/MWh. En el contexto del trabajo, trabajaremos con precios en €/kWh, por lo que habrá que dividir esas cantidades entre 1000.

Desafortunadamente, los datos del fichero no están ordenados cronológicamente, sino por el identificador del indicador que hace referencia al precio horario de la electricidad en la tarifa PVPC, que cambió en junio de 2021.

Se muestran a continuación las primeras y últimas líneas del fichero «tarifas-2021-ene-nov.csv», en las que se muestra en negrita la información relevante:

```

id;name;geoid;geoname;value;datetime
1001;Término de facturación [...] del PVPC 2.0TD;8741;Península;116.33;2021-06-01T00:00:00+02:00
1001;Término de facturación [...] del PVPC 2.0TD;8741;Península;115.95;2021-06-01T01:00:00+02:00
1001;Término de facturación [...] del PVPC 2.0TD;8741;Península;114.89;2021-06-01T02:00:00+02:00
...
1013;Término de facturación [...] del PVPC peaje por defecto;;;162.79;2021-05-31T21:00:00+02:00
1013;Término de facturación [...] del PVPC peaje por defecto;;;163.09;2021-05-31T22:00:00+02:00
1013;Término de facturación [...] del PVPC peaje por defecto;;;163.91;2021-05-31T23:00:00+02:00

```

⁷<https://www.iso.org/standard/70907.html>

Así, la segunda línea del fichero informa de que el 1 de junio de 2021, entre las 0:00 y las 0:59, el precio de la electricidad del PVPC en la península fue de 116,33 €/MWh, y la antepenúltima, de que el 31 de mayo de 2021, entre las 21:00 y las 21:59, el precio fue de 162,79 €/MWh.

7.2.2. Fichero de consumo eléctrico de un cliente

Se trata de conjuntos de ficheros que contienen datos sobre su consumo eléctrico que algunos de los clientes de Ineldo S.L. han facilitado a la empresa para ser asesorados. Cada fichero ha sido descargado por cada cliente de la página web de su empresa distribuidora de energía y contienen sus datos de consumo en cada hora de cada día de un mes concreto. Los ficheros siguen el siguiente esquema de nombrado:

```
<nombre-fichero-consumo> ::= <nombre-cliente> "-" <año> "-" <mes> ".csv"
<nombre-cliente> ::= literal-cadena
<año> ::= literal-entero
<mes> ::= literal-entero
```

El mes aparece siempre como un literal de dos dígitos, con un 0 a la izquierda si es preciso.

Puedes encontrar estos ficheros en el directorio «datos/» del repositorio de GitHub. A modo de ejemplo, el fichero «a-2021-10.csv» contiene los datos de consumo del mes de octubre de 2021 de la cliente «a». En estos momentos, se dispone de los datos de dos clientes, aunque se espera disponer de más en un futuro. Para simplificar la interacción con el programa durante el desarrollo del mismo, vamos a identificar a estos clientes como la cliente «a» y el cliente «b», aunque está previsto identificarlos a través de sus apellidos reales (que en los casos que nos ocupan eran «Aguilar» y «Blanco»). También por el momento, solo se dispone de datos de 2021 entre enero y noviembre, aunque la idea es que los clientes actuales y futuros sigan proporcionando datos de meses posteriores, conforme dispongan de ellos.

Todos los ficheros de consumo eléctrico siguen un formato compatible con el establecido por la CNMC para ser utilizados en su comparador de tarifas⁸ y en su simulador de facturas⁹, cuya estructura se muestra a continuación, expresada utilizando notación Backus-Naur:

```
<fichero-consumo> ::= <cabecera> {<datos-consumo-horario>}
<cabecera> ::= "CUPS;Fecha;Hora;AE_kwh;REAL/ESTIMADO" <fin-línea>
<datos-consumo-horario> ::= <cups> <sep> <fecha> <sep> <hora> <sep> <consumo>
                                <sep> <real-estimado> <fin-línea>
<cups> ::= literal-cadena
<fecha> ::= <día> "/" <mes> "/" <año>
<día> ::= literal-entero
<mes> ::= literal-entero
<año> ::= literal-entero
<hora> ::= literal-entero
<consumo> ::= literal-real
<real-estimado> ::= "R" | "E"
<fin-línea> ::= "\n"
<sep> ::= ","
```

⁸<https://comparador.cnmc.gob.es/>

⁹<https://comparador.cnmc.gob.es/facturaluz/inicio/>

Cada fichero comienza con una cabecera que ocupa una línea y le siguen líneas con información sobre el consumo horario de cada una de las horas de cada día del mes al que corresponde el fichero. Cada línea, además de los datos relevantes para este trabajo (<consumo>, <año>, <mes>, <día> y <hora>, que se muestran en negrita en el esquema sintáctico) contiene otro tipo de información en la que no estamos interesados, como el CUPS (Código Universal del Punto de Suministro) o si la lectura de consumo es real («R») o estimada («E»). Los datos del fichero están separados por el carácter punto y coma (';'), aunque han considerado la fecha como un único dato y tendremos que tener en cuenta que los datos <día>, <mes> y <año> están separados entre sí por el carácter '/ '.

En estos ficheros, **las horas no están comprendidas entre 0 y 23**, sino que están identificadas con números entre 1 y 24. **Los datos de consumo en cada hora están expresados en kWh**. En los ficheros que proporcionan las empresas distribuidoras, se utiliza la coma decimal como separador. En los ficheros proporcionados, se ha sustituido por el punto decimal para poder procesar el contenido de los mismos con más facilidad en el programa solicitado. También se han modificado en lo que respecta a los datos de los días 28 de marzo y 31 de octubre, para que reflejen datos de consumo horario de 24 horas (en lugar de las 23 y 25 horas que tuvieron realmente).

Se muestran a continuación las primeras y últimas líneas del fichero «a-2021-10.csv», en las que se muestra en negrita la información relevante:

```
CUPS;Fecha;Hora;AE_kWh;REAL/ESTIMADO
ES00313000000000001EX;01/10/2021;1;0.123;R
ES00313000000000001EX;01/10/2021;2;0.056;R
ES00313000000000001EX;01/10/2021;3;0.055;R
...
ES00313000000000001EX;31/10/2021;22;0.098;R
ES00313000000000001EX;31/10/2021;23;0.295;R
ES00313000000000001EX;31/10/2021;24;0.224;R
```

Así, la segunda línea del fichero informa de que el 1 de octubre de 2021, entre las **0:00** y las **0:59**, el consumo real registrado fue de 0,123 kWh, y la antepenúltima, de que el 31 de octubre de 2021, entre las **21:00** y las **21:59**, el consumo real fue de 0,098 kWh.

7.3. Requisitos

Se describe a continuación el comportamiento que debe tener el programa. Al final de este documento se incluyen las trazas completas de varias ejecuciones de ejemplo del programa (sección 7.8).

7.3.1. Petición de datos iniciales

El programa comienza solicitando al usuario el nombre de un cliente de Ineldo S.L. y el periodo de tiempo de 2021 expresado en meses que se quiere analizar.

Por ejemplo, si el usuario del programa quisiera analizar los datos del cliente «b» en el periodo entre junio y septiembre, ambos inclusive, la interacción producida con el programa sería la siguiente:

```
Escriba el nombre del usuario: b

Escriba el mes inicial y el final: 6 9
```

El programa debe asegurarse de que los meses introducidos pertenecen al rango de meses de 2021 para los que se dispone de datos (inicialmente, de enero a noviembre) y de que el mes inicial es anterior al mes final.

Si alguna o varias de esas condiciones no se cumplen, **el programa debe informar del problema al usuario y volver a solicitar el mes inicial y final, las veces que sea necesario hasta que todas ellas se satisfagan:**

```
Escriba el nombre del usuario: b

Escriba el mes inicial y el final: 9 6
El mes inicial tiene que ser menor o igual que el mes final.

Escriba el mes inicial y el final: 19 16
El mes inicial tiene que estar entre 1 y 11.
El mes final tiene que estar entre 1 y 11.
El mes inicial tiene que ser menor o igual que el mes final.

Escriba el mes inicial y el final: 16 19
El mes inicial tiene que estar entre 1 y 11.
El mes final tiene que estar entre 1 y 11.

Escriba el mes inicial y el final: 6 9
```

Una vez obtenidos el nombre de usuario y unos meses inicial y final válidos, el programa debe procesar los datos del fichero «tarifas-2021-ene-nov.csv» y los ficheros con los datos de consumo involucrados en la petición del usuario para estar en disposición de escribir el informe cuyo contenido se describe en la sección 7.3.3.

En el caso de las interacciones anteriores, los ficheros involucrados serían «tarifas-2021-ene-nov.csv», «b-2021-06.csv», «b-2021-07.csv», «b-2021-08.csv» y «b-2021-09.csv», todos ellos en el directorio «datos/».

En el caso de que no se pudiera leer de alguno de estos ficheros, el programa informará de ello y detendrá su ejecución:

```
Escriba el nombre del usuario: castro

Escriba el mes inicial y el final: 7 11

No se ha podido leer el fichero "datos/castro-2021-07.csv".
```

7.3.2. Petición del nombre del fichero en el que escribir los resultados

En el caso de que sí se haya podido procesar todos los ficheros involucrados, el programa continúa solicitando al usuario el nombre de un fichero en el que escribir el informe:


```
Escriba el nombre del usuario: a

Escriba el mes inicial y el final: 1 11

Escriba el nombre del fichero del informe
(presione solo ENTRAR para escribirlo en la pantalla): datos/informe.txt
Informe "datos/informe.txt" creado correctamente.
```


Si el usuario escribe el nombre de un fichero, el informe que se pide en la sección 7.3.3 se escribe en el fichero especificado por el usuario. El programa informa de si lo ha podido escribir correctamente o no, y termina.

No es responsabilidad del programa crear el fichero en un directorio concreto. Si, como en el ejemplo anterior, el usuario desea que el informe se genere en el directorio «datos/», es el propio usuario el que debe indicarlo añadiendo la ruta deseada al nombre del fichero.

Si el usuario opta por no escribir ningún nombre de fichero y simplemente pulsa la tecla «Entrar»  para continuar, el programa escribirá el informe directamente en la pantalla, sin necesidad de crear ningún fichero intermedio:

```
Escriba el nombre del usuario: a

Escriba el mes inicial y el final: 1 11

Escriba el nombre del fichero del informe
(presione solo ENTRAR para escribirlo en la pantalla): 

INFORME DEL CLIENTE "A" ENTRE LOS MESES 1 Y 11 DE 2021
-----
...
```

7.3.3. Contenido del informe

El informe que debe generarse una vez que el usuario del programa ha introducido los datos es un texto que se escribe en la pantalla o en un fichero de texto y consta de un encabezado y tres secciones, separadas por líneas en blanco, que se describen en las siguientes subsecciones.

Cabecera del informe

El informe comienza con una cabecera en mayúsculas la que se indica el nombre del cliente y los meses a los que se refiere el análisis. La cabecera tiene que tener el formato que se muestra a continuación:

```
INFORME DEL CLIENTE "A" ENTRE LOS MESES 1 Y 11 DE 2021
-----
```

Apartado 1: Tarifa PVPC en general

A continuación, deben aparecer dos líneas informando de:

- El día en el que el precio de la electricidad, en el conjunto de ese día, fue el más barato del periodo considerado, junto con ese precio medio.

Para obtener ese día y precio, deben considerarse los precios **medios** de las 24 horas de cada uno de los días del periodo considerado

- El día y hora del periodo considerado en el que se alcanzó el precio más caro del periodo considerado, junto con ese precio más caro.

Los precios deben expresarse en €/kWh, utilizando 5 decimales de precisión. El formato en el que se presenta la información de este apartado debe ser como el que se muestra a continuación:

El día completo más barato fue el 31-1-2021. Precio medio: 0.05773 €/kWh
La hora más cara tuvo lugar el 7-10-2021 a las 20:00. Precio: 0.41671 €/kWh

Apartado 2: Importe con la tarifa PVPC

A continuación, deben aparecer tres líneas informando de:

- **El importe del consumo eléctrico realizado por el cliente en el periodo considerado, de acuerdo con la tarifa del PVPC.**

Para obtener este importe, hay que multiplicar los consumos horarios del cliente en cada hora de cada día del periodo considerado (expresados en kWh) por el precio fijado por la tarifa PVPC para cada hora correspondiente (expresados en €/kWh) y sumarlos.

- **El importe mínimo correspondiente al consumo del cliente si, cada uno de los días del periodo considerado, hubiese sido capaz de realizar todo el consumo en la hora más barata de ese día.**

La situación descrita de concentrar todo el consumo de un día en tan solo una hora no es real, puesto que, por ejemplo, hay aparatos eléctricos (frigorífico, router, etc.) que deben funcionar todo el día. Sin embargo, el importe calculado bajo este supuesto proporciona una cota máxima al ahorro que puede producirse si se consiguiera trasladar parte del consumo eléctrico de horas más caras a horas más baratas (sin llegar a extremos como poner la lavadora a las 3 de la madrugada o levantarse para asar un pollo en el horno a las 6).

- **El porcentaje de ahorro que representaría poder concentrar todo el consumo en la hora más barata con respecto al importe real del consumo eléctrico según la tarifa PVPC, expresado como un tanto por ciento.**

Los precios deben expresarse en euros, utilizando 2 decimales de precisión. El formato en el que se presenta la información de este apartado debe ser como el que se muestra a continuación:

El importe del consumo eléctrico en el periodo considerado ha sido de 345.79 €.
El importe mínimo concentrando todo el consumo diario en la hora más barata habría sido de 273.90 € (un 20.79 % menor).

Apartado 3: Tarifas del mercado libre

Por último, en el informe debe aparecer una tabla con los importes del consumo eléctrico del cliente suponiendo que, en lugar de con la tarifa PVPC, hubiera estado acogido a una serie de contratos con comercializadoras del mercado libre.

Para el prototipo del programa que la empresa Ineldo S.L. nos ha encargado, se han seleccionado 8 tarifas de entre las 72 ofertas registradas por la CNMV. En concreto, se han elegido las dos aparentemente más baratas, las dos más caras, tres de la compañía comercializadora en el mercado libre del grupo correspondiente a la empresa distribuidora de electricidad en Aragón y la tarifa de precio fijo en mercado regulado que la comercializadora de referencia en Aragón está obligada a ofertar.

Las características de las tarifas en el mercado libre son muy variadas, pero las 8 tarifas seleccionadas tienen en común que, o bien ofrecen un precio fijo del kWh durante las 24 horas del día, o que ofrecen solo 3 precios distintos en función del tramo horario en los que se realice el consumo.

Estos tramos horarios son comunes para todas las ofertas de este tipo seleccionadas y sus denominaciones y características son los siguientes:

Punta: de 10:00 a 13:59 y de 18:00 a 21:59. Se asocia a los tramos horarios en los que la energía eléctrica es más cara.

Llano: de 8:00 a 9:59, de 14:00 a 17:59 y de 22:00 a 23:59. Se asocia a los periodos en los que el coste de la energía eléctrica es intermedio.

Valle: de 0:00 a 7:59. Asociado a los periodos en los que la energía eléctrica es más barata¹⁰.

Por simplicidad, los datos sobre los nombres y los precios de las 8 tarifas seleccionadas para el prototipo del programa se encuentran definidos en el propio código C++. En la sección 7.4 se proporciona más información al respecto.

El formato en el que se presenta la información de este apartado debe ser como el que se muestra a continuación:

| COSTE CON TARIFAS COMERCIALES | |
|-------------------------------|---|
| Coste | Nombre de la tarifa |
| ----- | |
| 389.18 € | Compañía NUF: Un precio 24h |
| 419.46 € | Compañía NUF: Tres precios |
| 411.05 € | Compañía END: UNA Light |
| 420.37 € | Compañía END: UNA 3 Periods |
| 518.49 € | Compañía END: Tiempo 3 Periods |
| 745.29 € | Compañía LOG: Alfa Hogar 2.0 |
| 722.94 € | Compañía LOG: Alfa Fijo Único Hogar |
| 396.34 € | Compañía E21: Precio Fijo en Mercado Regulado |

7.4. Material de partida

En el repositorio <https://github.com/prog1-eina/trabajo-2021-22> se encuentra el material de partida para la realización de este trabajo. El repositorio cuenta con la estructura de directorios y fiche-

¹⁰ Este periodo también incluye las 24 horas de los sábados, domingos y festivos nacionales. Sin embargo, simplificaremos el problema suponiendo que los tramos horarios son siempre tres, independientemente del día de la semana de que se trate.

ros necesaria para que el fichero «Makefile» y las tareas de compilación, ejecución y depuración de Visual Studio Code funcionen adecuadamente.

El área de trabajo de puede descargar de GitHub (botón [Code](#) [Download ZIP](#) de la web del repositorio), debiéndose descomprimir su contenido una vez concluida la descarga. Recuerda borrar el sufijo «-main» que añade GitHub al preparar el fichero comprimido para su descarga.

En el repositorio hay un directorio denominado «datos» donde se encuentran el fichero «tarifas-2021-ene-nov.csv», los ficheros desde «a-2021-01.csv» hasta «a-2021-11.csv» y los ficheros desde «b-2021-01.csv» hasta «b-2021-11.csv», correspondientes a los consumos de enero a noviembre de los clientes «a» y «b», que tendrá que utilizar el programa que se pide en este trabajo.

En el directorio «src» se encuentran los ficheros correspondientes a los seis módulos en los que estará repartido el código de este trabajo:

1. El módulo principal se ubica en el fichero «main.cpp». En este módulo se encuentra un esqueleto de las funciones `main` y `escribirInforme`. Esta última está pensada para escribir un informe tanto en cout como en un flujo de la clase `ofstream` que haya sido declarado y asociado a un fichero externo con anterioridad.

Este módulo puede hacer uso de la funciones de los otros dos módulos que han de desarrollarse y pueden definirse en él todas aquellas funciones adicionales que se estimen convenientes.

2. El módulo «fecha» ya está implementado y define un tipo registro denominado `Fecha`, junto con una función denominada `mostrar` que permiten escribir una fecha en la pantalla o en un flujo de la clase `ofstream`.

También define otra función, denominada `diasTranscurridos` que, dadas dos fechas, devuelve el número de días que han transcurrido entre ellas.

3. El módulo «gasto-diario» define un tipo registro denominado `GastoDiario`, pensado para reflejar los 24 datos horarios de consumo y precio de **un determinado día**. Debe completarse la definición del tipo `GastoDiario` con los campos necesarios para que los datos de este tipo puedan almacenar la información solicitada.

En el fichero de interfaz de este módulo, «gasto-diario.hpp» se declaran también varias funciones que permiten obtener, a partir de la información de un registro de tipo `GastoDiario`, datos que resumen aspectos concretos de la información de ese registro:

- `unsigned` `horaMasCara(const GastoDiario& gasto);`
Debe devolver la hora de mayor precio de las 24 horas del día al que hace referencia el parámetro `gasto`.
- `double` `costeMedio(const GastoDiario& gasto);`
Debe devolver el precio medio de la electricidad del día al que hace referencia el parámetro `gasto`, promediando los 24 precios horarios de ese día.
- `double` `costeDiario(const GastoDiario& gasto);`
Debe devolver el importe en euros del coste de la energía eléctrica del día al que hace referencia el parámetro `gasto`, de acuerdo con los 24 datos de consumo y precios horarios del propio registro `gasto`.
- `double` `costeDiarioMinimo(const GastoDiario& gasto);`
Debe devolver el importe en euros del coste de la energía eléctrica del día al que hace referencia el parámetro `gasto`, suponiendo que todo el consumo eléctrico reflejado en dicho parámetro hubiera tenido lugar en la hora con el precio más barato del registro `gasto`.

Estas funciones deberán ser implementadas en el fichero «gasto-diario.cpp».

4. El módulo «vector-gastos» define un conjunto de funciones que serán útiles para la escritura del informe por parte del programa solicitado. Estas funciones tienen como parámetros de entrada un vector de datos de tipo **GastoDiario** y un número natural que indica el número de componentes que tiene ese vector. Cuando se invoque a estas funciones, ese número de componentes coincidirá con el número de días contenidos en el intervalo de tiempo indicado por el usuario en el programa interactivo.

Las funciones declaradas en el fichero de interfaz «vector-gastos.hpp» y que deberán ser implementadas en el fichero «vector-gastos.cpp» son:

- **void** diaMasBarato(**const GastoDiario** regsDiarios[], **const unsigned** numRegs, **Fecha&** dia, **double&** precioMedioMinimo);

Debe asignar a sus parámetros de salida el día de menor precio promedio de entre las numRegs componentes del vector regsDiarios, así como el valor de dicho precio promedio mínimo. Esta función obtendrá parte de la información necesaria para cumplimentar parte del apartado 1 del informe y en su implementación podrá hacer uso de la función `costeMedio` del módulo «gasto-diario».

- **void** horaMasCara(**const GastoDiario** regsDiarios[], **const unsigned** numRegs, **Fecha&** dia, **unsigned&** hora, **double&** precioMaximo);

Debe asignar a sus parámetros de salida el día y la hora de mayor precio de entre las numRegs componentes del vector regsDiarios, así como el valor de dicho precio mayor. Esta función obtendrá parte de la información necesaria para cumplimentar parte del apartado 1 del informe y en su implementación podrá hacer uso de la función del mismo nombre del módulo «gasto-diario».

- **double** costeTerminoVariable(**const GastoDiario** regsDiarios[], **const unsigned** numRegs);

Debe devolver el coste en euros de la energía eléctrica consumida según los datos de las numRegs componentes del vector regsDiarios. Esta función obtendrá parte de la información necesaria para cumplimentar parte del apartado 2 del informe y en su implementación podrá hacer uso de la función `costeDiario` del módulo «gasto-diario».

- **double** costeMinimoPosible(**const GastoDiario** regsDiarios[], **const unsigned** numRegs);

Debe devolver el coste en euros de la energía eléctrica consumida según los datos de las numRegs componentes del vector regsDiarios suponiendo que todo el consumo de cada uno de los días se ha producido precisamente en la hora más barata de ese día. Esta función obtendrá parte de la información necesaria para cumplimentar parte del apartado 2 del informe y en su implementación podrá hacer uso de la función `costeDiarioMinimo` del módulo «gasto-diario».

5. El módulo «fichs-electricos.hpp» define un conjunto de funciones que serán útiles para leer los datos de los ficheros de precios y de consumos en un vector de datos de tipo **GastoDiario**:

- **bool** leerPrecioHorario(**istream&** f, **Fecha&** fecha, **unsigned&** hora, **double&** precio);

Esta función sirve para extraer línea a línea los datos de precios del fichero de precios horarios. Dado un flujo de entrada f asociado con dicho fichero y estando en disposición de leer desde el principio de una línea que contiene información sobre un precio horario, esta función extrae del flujo esa línea (y solo esa) y asigna a los parámetros fecha, hora y precio los datos sobre la fecha, hora y precio horario correspondientes a esa línea. La función devolverá **true** si no se han terminado los datos del fichero en el intento de lectura y **false** en caso contrario.

- `bool leerConsumoHorario(istream& f, Fecha& fecha, unsigned& hora, double& consumo);`

Esta función es análoga a la anterior, pero con consumos: sirve para extraer línea a línea los datos de consumo de un fichero de consumos horarios. Dado un flujo de entrada `f` asociado con uno de estos ficheros y estando en disposición de leer desde el principio de una línea que contiene información sobre un consumo horario, esta función extrae del flujo esa línea (y solo esa) y asigna a los parámetros `fecha`, `hora` y `consumo` los datos sobre la fecha, hora y consumo horario correspondientes a esa línea. La función devolverá `true` si no se han terminado los datos del fichero en el intento de lectura y `false` en caso contrario.

- `bool leerPrecios(const string nombreFichero, const unsigned mesInicial, const unsigned mesFinal, GastoDiario registros[]);`

Esta función sirve para extraer la información del fichero de precios correspondientes a todas las horas de fechas comprendidas entre `mesInicial` y `mesFinal` y almacenarla en el vector `registros`. El nombre del fichero de precios viene dado por el parámetro `nombreFichero`. La función devolverá `true` si se ha podido procesar el fichero de nombre `nombreFichero` correctamente y `false` en caso contrario.

Los registros del vector `registros` deben quedar ordenados por orden cronológico. Para ello, puede ser útil la función `diasTranscurridos` que ya se ofrece implementada en el módulo «fecha»: el número de días transcurridos entre el primer día del `mesInicial` y una fecha concreta coincide con el índice de la componente de tipo `GastoDiario` correspondiente a esa fecha en el vector `registros`. Por lo tanto, esta función se puede implementar como una sucesión de invocaciones a la función `leerPrecioHorario`, en la que tras cada lectura de un precio horario, el dato correspondiente se ubica en la componente adecuada del vector.

Esta función no incluye como parámetro ni de entrada ni de salida el número de componentes del vector `registros`, ya que este puede ser calculado a partir de los datos `mesInicial` y `mesFinal` utilizando también la función `diasTranscurridos`.

- `bool leerConsumos(const string nombreCliente, const unsigned mesInicial, const unsigned mesFinal, GastoDiario registros[]);`

Esta función sirve para extraer la información de los ficheros de consumos correspondientes a todas las horas de fechas comprendidas entre `mesInicial` y `mesFinal` del cliente de nombre `cliente` y almacenarla en el vector `registros`. La función devolverá `true` si se han podido procesar correctamente todos los ficheros de consumo involucrados y `false` en caso contrario.

El nombre de los ficheros de consumo deberá construirse en esta función. Para ello, puede ser útil conocer la función `to_string`¹¹: dado un dato de tipo entero, esta función devuelve una cadena de caracteres con la secuencia de dígitos que lo representa.

Igual que en el caso de la función `leerPrecios`, los registros del vector `registros` deben quedar ordenados por orden cronológico y será también de utilidad la función `diasTranscurridos`. Esta función también se puede implementar de forma análoga a `leerPrecios`: como una sucesión de invocaciones a la función `leerConsumoHorario`, en la que tras cada lectura de un consumo horario, el dato correspondiente se ubica en la componente adecuada del vector.

Esta función **tampoco incluye como parámetro ni de entrada ni de salida el número de componentes del vector** `registros`, ya que este puede ser calculado a partir de los datos `mesInicial` y `mesFinal` utilizando la función `diasTranscurridos`.

En las postcondiciones de estas dos últimas funciones se advierte claramente de que la función `leerPrecios` no modifica los datos de consumos de las componentes de su parámetro `registros`

¹¹http://www.cplusplus.com/reference/string/to_string/

y de que la función leerConsumos tampoco modifica los datos de precios de las componentes del suyo. Esto es así porque, cuando sean invocadas, ambas trabajarán con el mismo argumento para sus respectivos parámetros registros: la invocación a una de ellas almacenará los precios o consumos horarios en un vector y, posteriormente, la otra hará lo propio con el otro tipo de información en el mismo vector.

La correcta implementación de las funciones de este módulo es crítica, puesto que ninguno de los apartados del informe podrán contener resultados correctos si estas funciones no están correctamente implementadas. Este hecho debe ser tenido en cuenta a la hora de planificar y ejecutar el trabajo.

6. El módulo «tarifas-comerciales» define un tipo registro denominado TarifaPlanaTramos, declarado para representar los precios de los tres tramos (punta, llano y valle) de una tarifa comercial que ofrece un precio para cada uno de esos periodos. Este registro también sirve para representar tarifas en las que el precio por hora es constante durante todo el día, sin considerar tramos horarios: basta con hacer que los tres campos correspondientes a los precios en cada tramo tengan el mismo valor.

En este módulo se ha definido un vector de 8 componentes de tipo TarifaPlanaTramos, en las que cada una representa una de las 8 ofertas con las que la empresa Ineldo S.L. quiere trabajar en el prototipo del programa de análisis.

En el fichero de interfaz de este módulo, «tarifas-comerciales.hpp», se han declarado también dos funciones (costeDiarioTarifaPlanaTramos y costeTarifaPlanaTramos) que permiten obtener, respectivamente, a partir de la información de un registro de tipo **GastoDiario** o de un vector de tipo **GastoDiario**, el importe diario o el total de un periodo del consumo de un cliente utilizando una determinada tarifa.

En cualquiera de los módulos (preferiblemente en los ficheros de implementación) pueden añadirse todas las funciones auxiliares que se estime oportuno.

7.5. Equipos de trabajo

La formación de los equipos de dos estudiantes para la realización de este trabajo será libre y se realizará a través de la tarea de Moodle «Formación de grupos para el trabajo obligatorio»¹². Los equipos podrán estar integrados por cualquier pareja de estudiantes matriculados en la asignatura, pudiendo ser de distintos grupos de teoría o prácticas. Quien no encuentre pareja para apuntarse en la tarea, podrá inscribirse en un equipo de forma individual y esperar a que alguien en la misma situación se inscriba en su grupo, o bien inscribirse en un equipo en el que ya haya un estudiante, para completarlo. Cuando os inscribáis en un equipo, debéis ser conscientes de que adquirís un compromiso personal y académico con el otro miembro del equipo en cuanto a la realización, finalización y entrega del trabajo.

En caso de desavenencias entre los miembros de un equipo, se ofrece la posibilidad de modificar la composición de los mismos en cualquier momento (dentro de la disponibilidad de huecos que pueda haber), con la fecha límite correspondiente al último día de clases de este cuatrimestre (13 de enero de 2022). A partir de ese momento, la composición de los grupos quedará fijada de forma definitiva.

El programa que se pide realizar tiene una carga de trabajo suficiente como para que sea objeto de una cuidadosa planificación por parte de los miembros de cada equipo, que va más allá de repartirse distintas funcionalidades para ser desarrolladas independientemente y luego integradas en una entrega final. **Lo que se espera es que realmente se trabaje en equipo, con comunicación frecuente y reuniones para la coordinación del trabajo y la implementación del mismo.**

¹²<https://moodle.unizar.es/add/mod/choicegroup/view.php?id=2504511>

La forma de colaboración que se sugiere es similar a la que en metodologías ágiles se denomina *programación por pares*¹³ (*pair programming* en inglés). Aplicando esta técnica, dos programadores trabajan simultáneamente en la escritura del código de un programa de forma conjunta, a la vez que planean y reflexionan sobre el trabajo que realizan, aclaran sus ideas y evalúan soluciones alternativas, dando como resultado de mejor calidad¹⁴.

Habitualmente, se definen dos papeles:

1. *Piloto* o *controlador* (*driver* en inglés), que es la persona que utiliza el teclado en un momento dado y, por lo tanto, escribe el código, concentrada en las líneas de código que está escribiendo, mientras comenta al otro programador lo que está haciendo.
2. *Navegador* u *observador* (*navigator* en inglés), que es la persona que se mantiene en una posición observadora mientras la otra está tecleando. Se encarga de revisar el código que escribe su compañero mientras lo escribe, le guía y comparte ideas. Se mantiene alerta ante problemas o errores sintácticos o de ejecución que puedan estar introduciéndose en el código y va pensando en los siguientes pasos a realizar.

Se recomienda alternar estos dos roles con regularidad.

La técnica de programación por pares puede aplicarse de forma presencial, compartiendo un único equipo, o de forma remota. La extensión Live Share¹⁵ de Visual Studio Code es una posibilidad. Requiere de una cuenta en GitHub¹⁶ o de Microsoft para trabajar.

También es posible aplicarla de forma remota reuniéndose en una sala de Google Meet. El piloto escribe el código mientras comparte su pantalla y el navegador observa, guía y comparte sus ideas. El cambio de roles implica intercambiar primero el código (a través del correo electrónico, una carpeta compartida en Google Drive o incluso desde la tarea de entrega del trabajo en Moodle) y, después, que el nuevo piloto comparta su pantalla.

7.6. Recomendaciones y restricciones de diseño

Modularidad del diseño. El programa debe diseñarse de forma que esté integrado por un conjunto de funciones repartidas en los módulos que se suministran en el código de partida. Estas funciones deberían resolver problemas acotados y concretos, siempre que sea posible. Se recuerda que copiar y pegar código no es una forma adecuada de reutilizar código: ante situaciones en las que la solución a parte de un problema es similar a parte de otro, debe definirse una función, adecuadamente parametrizada, que permita resolver los problemas iniciales haciendo las invocaciones adecuadas a la función creada.

El número de líneas de código de cada función debe ser reducido. Conviene facilitar la comprensión del código de aquellas funciones que revistan cierta complejidad incorporando, con medida, comentarios que aclaren su comportamiento y eviten tener que hacer un seguimiento detallado del código para comprender su comportamiento. En cualquier caso, se recuerda que la mejor documentación del código consiste en utilizar identificadores adecuados y significativos y utilizar funciones que resuelvan problemas concretos siempre que sea posible.

Documentación. Las primeras líneas de los ficheros que forman el programa estarán dedicadas a un comentario en el que figuren los nombres y apellidos de los componentes del equipo.

¹³<https://martinfowler.com/bliki/PairProgramming.html>

¹⁴<https://martinfowler.com/articles/on-pair-programming.html>

¹⁵<https://docs.microsoft.com/en-us/visualstudio/liveshare/>

¹⁶<https://github.com/>

Así mismo, cada función debe contar con su propia especificación mediante:

1. Una precondition que describa con precisión las condiciones que, en su caso, deben satisfacer sus datos de entrada.
2. Una postcondición que describa con precisión las condiciones que deben satisfacer sus datos de salida o resultados.

Legibilidad. La edición del programa fuente debe facilitar su lectura y comprensión. Para ello se deberán tener en cuenta, entre otros, los siguientes aspectos:

1. Elección de identificadores significativos.
2. Alineación y sangrado de las declaraciones y del código de cada algoritmo que faciliten la comprensión de su estructura.
3. Evitar líneas de longitud excesiva que dificulten su lectura en pantalla o su impresión. No conviene que la longitud de ninguna línea sobrepase los 120 caracteres.

Modificabilidad. El diseño del programa debe facilitar los cambios y modificaciones que se realicen durante la explotación del programa. En concreto se deberá velar por facilitar la realización, en su caso, de los siguientes cambios:

1. La incorporación de datos sobre consumo de nuevos clientes, que se realizaría añadiendo los ficheros de consumos de los nuevos clientes y que seguirían la sintaxis de la regla <fichero-consumo> definida en la sección 7.2.
2. La incorporación de datos de precios y consumo del mes de diciembre. En estos momentos, se tiene la certeza de que a partir de enero estarán disponibles los datos de precios del año 2021 completo y que los clientes «a» y «b» proporcionarán sus datos de consumo. Las modificaciones en el programa necesarias para poder incorporar estos datos deberían consistir únicamente en modificar los valores de las constantes MES_INICIAL o MES_FINAL del fichero «gasto-diario.hpp».

7.7. Entrega y evaluación del trabajo

Antes del ~~sábado 15 de enero de 2022 a las 18:00~~ jueves 27 de enero de 2022 a las 23:59, se deben haber subido a Moodle los siguientes ficheros:

- «fichs-electricos.cpp»
- «fichs-electricos.hpp»
- «gasto-diario.cpp»
- «gasto-diario.hpp»
- «main.cpp»
- «tarifas-comerciales.cpp»
- «tarifas-comerciales.hpp»
- «vector-gastos.cpp»
- «vector-gastos.hpp»

El módulo «fecha» no debe ser modificado, por lo que no se deben entregar los ficheros «fecha.hpp» y «fecha.cpp».

Cada uno de los trabajos realizados por cada equipo será evaluado y calificado sobre 10 puntos. La calificación final de la primera convocatoria tiene en cuenta la nota de este trabajo en un 15 %.

Para posibilitar una compilación automática, deben entregarse todos los ficheros indicados en el apartado anterior, incluso si en la descomposición modular realizada, alguno de ellos no hubiera sido utilizado. En este caso, pueden entregarse ficheros sin contenido, pero deben ser subidos a Moodle igualmente. El código fuente entregado debe poderse compilar utilizando el fichero «Makefile» original suministrado en el repositorio del material de partida (es decir, no se debe modificar el «Makefile» proporcionado).

Cada uno de los ficheros entregados estará encabezado por un comentario en el que consten el nombre y apellidos de los autores del trabajo.

No se admitirán trabajos presentados mediante otros medios ni presentados fuera de plazo.

El trabajo que se propone debe ser realizado por cada equipo con total independencia de los demás equipos. Debe evitarse que miembros de otros equipos tengan acceso a los documentos de trabajo y programas desarrollados por el equipo. **Se hará un análisis de plagios entre trabajos.** La copia o coincidencia manifiesta de una parte o de la totalidad de un programa con el presentado por otro equipo conllevará una calificación de un cero a todos los miembros de los equipos implicados.

Al corregir, se compilará el código de los ficheros entregados. La existencia de errores sintácticos en los mismos supondrá un cero en la calificación del trabajo.

Para calificar el trabajo se tendrán en cuenta los siguientes aspectos:

- **Fidelidad del comportamiento del programa a lo especificado y descrito en este documento** (50 % de la calificación del trabajo).

Se admitirán entregas de programas que no satisfagan la totalidad de los requisitos planteados en este enunciado, siempre que el programa pueda ser compilado correctamente y que la ejecución del programa se produzca sin errores. En todo caso, los distintos requisitos planteados en este enunciado tendrán pesos proporcionales a la dificultad de los mismos a la hora de establecer la calificación, y la no implementación de algunos requisitos también repercutirá en las calificaciones de los siguientes apartados.

- **Calidad del diseño algorítmico del código (datos y funciones)**, atendiendo los diferentes criterios de calidad explicados en la asignatura (diseño descendente, eficiencia, reusabilidad del

código, etc.) (35 % de la calificación del trabajo).

- **Presentación y legibilidad del código**, teniendo muy presentes las reglas publicadas en la *Guía de estilo para programar en C++*, accesible desde Moodle¹⁷ (15 % de la calificación del trabajo).

El código C++ del programa desarrollado por el equipo debe ser conservado por cada estudiante durante todo el curso.


7.8. Trazas de varias ejecuciones de ejemplo del programa

7.8.1. Traza 1

Escriba el nombre del cliente: a

Escriba el mes inicial y el final: 1 11

Escriba el nombre del fichero del informe

(presione solo ENTRAR para escribirlo en la pantalla): 

INFORME DEL CLIENTE "A" ENTRE LOS MESES 1 Y 11 DE 2021

El día completo más barato fue el 31-1-2021. Precio medio: 0.05773 €/kWh

La hora más cara tuvo lugar el 7-10-2021 a las 20:00. Precio: 0.41671 €/kWh

El importe del consumo eléctrico en el periodo considerado ha sido de 345.79 €.

El importe mínimo concentrando todo el consumo diario en la hora más barata
habría sido de 273.90 € (un 20.79 % menor)

COSTE CON TARIFAS COMERCIALES

| Coste | Nombre de la tarifa |
|----------|---|
| 389.18 € | Compañía NUF: Un precio 24h |
| 419.46 € | Compañía NUF: Tres precios |
| 411.05 € | Compañía END: UNA Light |
| 420.37 € | Compañía END: UNA 3 Periods |
| 518.49 € | Compañía END: Tiempo 3 Periods |
| 745.29 € | Compañía LOG: Alfa Hogar 2.0 |
| 722.94 € | Compañía LOG: Alfa Fijo Único Hogar |
| 396.34 € | Compañía E21: Precio Fijo en Mercado Regulado |

¹⁷<https://moodle.unizar.es/add/mod/resource/view.php?id=2504550>

7.8.2. Traza 2

Escriba el nombre del cliente: b

Escriba el mes inicial y el final: 11 11

Escriba el nombre del fichero del informe

(presione solo ENTRAR para escribirlo en la pantalla): informe-b-noviembre.txt

Informe "informe-b-noviembre.txt" creado correctamente.

Como resultado de la ejecución anterior, se habrá creado un fichero denominado «informe-b-noviembre.txt» en el directorio de ejecución del programa (el directorio «trabajo-2021-22/») con el siguiente contenido:

INFORME DEL CLIENTE "B" ENTRE LOS MESES 11 Y 11 DE 2021

El día completo más barato fue el 1-11-2021. Precio medio: 0.12600 €/kWh

La hora más cara tuvo lugar el 30-11-2021 a las 18:00. Precio: 0.40377 €/kWh

El importe del consumo eléctrico en el periodo considerado ha sido de 156.49 €.

El importe mínimo concentrando todo el consumo diario en la hora más barata

habría sido de 125.37 € (un 19.89 % menor)

COSTE CON TARIFAS COMERCIALES

| Coste | Nombre de la tarifa |
|----------|---|
| 95.34 € | Compañía NUF: Un precio 24h |
| 103.80 € | Compañía NUF: Tres precios |
| 100.69 € | Compañía END: UNA Light |
| 103.16 € | Compañía END: UNA 3 Periods |
| 127.20 € | Compañía END: Tiempo 3 Periods |
| 182.86 € | Compañía LOG: Alfa Hogar 2.0 |
| 177.10 € | Compañía LOG: Alfa Fijo Único Hogar |
| 97.28 € | Compañía E21: Precio Fijo en Mercado Regulado |

7.8.3. Traza 3

Escriba el nombre del cliente: f

Escriba el mes inicial y el final: 6 11

No se ha podido leer el fichero "datos/f-2021-06.csv".

7.8.4. Traza 4

Escriba el nombre del cliente: b

Escriba el mes inicial y el final: 9 6

El mes inicial tiene que ser menor o igual que el mes final.

Escriba el mes inicial y el final: 19 16

El mes inicial tiene que estar entre 1 y 11.

El mes final tiene que estar entre 1 y 11.

El mes inicial tiene que ser menor o igual que el mes final.

Escriba el mes inicial y el final: 16 19

El mes inicial tiene que estar entre 1 y 11.

El mes final tiene que estar entre 1 y 11.

Escriba el mes inicial y el final: 6 9

Escriba el nombre del fichero del informe

(presione solo ENTRAR para escribirlo en la pantalla):

INFORME DEL CLIENTE "B" ENTRE LOS MESES 6 Y 9 DE 2021

El día completo más barato fue el 20-6-2021. Precio medio: 0.07690 €/kWh

La hora más cara tuvo lugar el 15-9-2021 a las 20:00. Precio: 0.35404 €/kWh

El importe del consumo eléctrico en el periodo considerado ha sido de 144.13 €.

El importe mínimo concentrando todo el consumo diario en la hora más barata
habría sido de 96.49 € (un 33.05 % menor)

COSTE CON TARIFAS COMERCIALES

| Coste | Nombre de la tarifa |
|----------|---|
| 127.30 € | Compañía NUF: Un precio 24h |
| 142.82 € | Compañía NUF: Tres precios |
| 134.46 € | Compañía END: UNA Light |
| 139.73 € | Compañía END: UNA 3 Periods |
| 171.21 € | Compañía END: Tiempo 3 Periods |
| 246.91 € | Compañía LOG: Alfa Hogar 2.0 |
| 236.48 € | Compañía LOG: Alfa Fijo Único Hogar |
| 131.88 € | Compañía E21: Precio Fijo en Mercado Regulado |

7.8.5. Traza 5

Escriba el nombre del cliente: a

Escriba el mes inicial y el final: 1 5

Escriba el nombre del fichero del informe

(presione solo ENTRAR para escribirlo en la pantalla): datos/inf-a-ene-may.txt

Informe "datos/inf-a-ene-may.txt" creado correctamente.

Como resultado de la ejecución anterior, se habrá creado un fichero denominado «inf-a-ene-may.txt» en el directorio «datos/») con el siguiente contenido:

INFORME DEL CLIENTE "A" ENTRE LOS MESES 1 Y 5 DE 2021

El día completo más barato fue el 31-1-2021. Precio medio: 0.05773 €/kWh

La hora más cara tuvo lugar el 9-1-2021 a las 20:00. Precio: 0.20272 €/kWh

El importe del consumo eléctrico en el periodo considerado ha sido de 219.40 €.

El importe mínimo concentrando todo el consumo diario en la hora más barata
habría sido de 181.62 € (un 17.22 % menor)

COSTE CON TARIFAS COMERCIALES

| Coste | Nombre de la tarifa |
|-------|---------------------|
|-------|---------------------|

| | |
|----------|---|
| 288.38 € | Compañía NUF: Un precio 24h |
| 310.08 € | Compañía NUF: Tres precios |
| 304.58 € | Compañía END: UNA Light |
| 310.53 € | Compañía END: UNA 3 Periods |
| 383.65 € | Compañía END: Tiempo 3 Periods |
| 551.01 € | Compañía LOG: Alfa Hogar 2.0 |
| 535.69 € | Compañía LOG: Alfa Fijo Único Hogar |
| 292.75 € | Compañía E21: Precio Fijo en Mercado Regulado |

7.8.6. Traza 6

Escriba el nombre del cliente: b

Escriba el mes inicial y el final: 6 11

Escriba el nombre del fichero del informe

(presione solo ENTRAR para escribirlo en la pantalla): b@jun-nov.txt

Informe "b@jun-nov.txt" creado correctamente.

Como resultado de la ejecución anterior, se habrá creado un fichero denominado «b@jun-nov.txt» en el directorio de ejecución del programa con el siguiente contenido:

INFORME DEL CLIENTE "B" ENTRE LOS MESES 6 Y 11 DE 2021

El día completo más barato fue el 20-6-2021. Precio medio: 0.07690 €/kWh

La hora más cara tuvo lugar el 7-10-2021 a las 20:00. Precio: 0.41671 €/kWh

El importe del consumo eléctrico en el periodo considerado ha sido de 388.51 €.

El importe mínimo concentrando todo el consumo diario en la hora más barata
habría sido de 290.05 € (un 25.34 % menor)

COSTE CON TARIFAS COMERCIALES

| Coste | Nombre de la tarifa |
|-------|---------------------|
|-------|---------------------|

| | |
|----------|---|
| 275.16 € | Compañía NUF: Un precio 24h |
| 305.19 € | Compañía NUF: Tres precios |
| 290.61 € | Compañía END: UNA Light |
| 300.30 € | Compañía END: UNA 3 Periods |
| 368.90 € | Compañía END: Tiempo 3 Periods |
| 531.33 € | Compañía LOG: Alfa Hogar 2.0 |
| 511.13 € | Compañía LOG: Alfa Fijo Único Hogar |
| 283.34 € | Compañía E21: Precio Fijo en Mercado Regulado |

7.8.7. Traza 7

Escriba el nombre del cliente: a

Escriba el mes inicial y el final: 1 11

Escriba el nombre del fichero del informe

(presione solo ENTRAR para escribirlo en la pantalla): nombre>fichero>no>valido.txt

No se ha podido escribir en el fichero "nombre>fichero>no>valido.txt".

Como resultado de la ejecución anterior, no se habrá creado ningún fichero.