

# Práctica Markov

INTELIGENCIA ARTIFICIAL

**Ángela Serrano Casas**

NIA: 100475053

Correo: 100475053@alumnos.uc3m.es

Grupo: 80

Titulación: Ingeniería informática

**Natalia Rodríguez Navarro:**

NIA: 100471976

Correo: 100471976@alumnos.uc3m.es

Grupo: 80

Titulación: Ingeniería informática

# Índice

INTRODUCCIÓN Y OBJETIVOS	2
RESUMEN EJECUTIVO	3
DESCRIPCIÓN FORMAL DEL MODELO MDP	4
ANÁLISIS DETALLADO DEL MODELO DE COSTES	5
POLÍTICA ÓPTIMA PARA LOS COSTES ELEGIDOS	7
FASES DEL PROYECTO	10
Diseño	10
Implementación	12
Pruebas	13
PRESUPUESTO	16
CONCLUSIONES	18

---

## INTRODUCCIÓN Y OBJETIVOS

El propósito principal del presente documento es mostrar el proceso atravesado a lo largo de estas últimas semanas para el desarrollo de la práctica final de la asignatura.

Dicha práctica busca implementar la toma de decisiones de un sistema de calefacción. Las posibles acciones que puede tomar son encenderse o apagarse. Estas decisiones se toman cada media hora, buscando siempre alcanzar la temperatura deseada, en nuestro caso establecida a 22 grados pero ésta puede ser modificada. La temperatura de la que se parte no se debe especificar, es por ello que para cada posible temperatura que puede detectar el termostato se debe devolver la acción más recomendable para alcanzar la temperatura ideal.

Para el desarrollo de esta práctica hemos tenido en cuenta distintos puntos claves que se exponen a lo largo del documento. En primer lugar, se dará un resumen ejecutivo del proyecto (apartado [RESUMEN EJECUTIVO](#)). Daremos a conocer los puntos claves del proyecto que son necesarios para el entendimiento del mismo. Explicaremos en segundo lugar el marco teórico sobre el que nos movemos. Aquella parte del temario de la asignatura que hemos utilizado para poner en práctica este proyecto (capítulo [DESCRIPCIÓN FORMAL DEL MODELO MDP](#)). Posteriormente, se justificarán los valores utilizados como costes de las acciones del sistema de calefacción: apagarse o encenderse ([ANÁLISIS DETALLADO DEL MODELO DE COSTES](#)). Se incluye también la respuesta del sistema ante estos costes en cada uno de los estados (sección [POLÍTICA ÓPTIMA PARA LOS COSTES ELEGIDOS](#)). A continuación se muestra el último punto clave del proyecto. Este muestra el proceso de desarrollo del código implementado. Las fases principales para conseguir la funcionalidad pedida y una pequeña batería de pruebas para ratificar su buen funcionamiento ([FASES DEL PROYECTO](#)). Se incluye también el presupuesto que debería tener como mínimo el proyecto para poder desarrollarse en el tiempo y forma que ha sido desarrollado ([PRESUPUESTO](#)). Finalmente, se incluyen los puntos a resaltar del presente documento en el apartado de [CONCLUSIONES](#): comentarios técnicos acerca del proyecto y las opiniones personales que se han generado tras la finalización de este.

## RESUMEN EJECUTIVO

El proyecto en cuestión trata sobre la implementación en el lenguaje de programación [Python](#) de un Modelo de Decisión de Markov. Éste nos permitirá obtener qué acciones ejecutar para llegar a cierta meta con alta probabilidad y menor coste.

En este caso, realizamos un análisis para atacar el problema de toma de decisiones en condiciones de incertidumbre respecto a una calefacción que tiene que escoger entre **encenderse** o **apagarse** para llegar a la temperatura deseada e indicada por el usuario. La temperatura deseada y temperaturas iniciales posibles que valoraremos serán entre **16 y 25 grados con saltos de 0.5 en 0.5**.

Al ser un MDP definimos una política que ayudará a decidir qué acción ejecutar para cualquier estado y que garantiza llegar a una meta. Esta política óptima es más que un plan, ya que tolera fallos. Entonces, el proceso a implementar es markoviano: el resultado “s’” de aplicar la acción “a” sólo depende del estado “s”. Por esta razón, se calcula la política óptima para todos los estados posibles.

En resumen, nuestra implementación contempla los estados (temperaturas) y las acciones (on/ off) descritas, junto con unos **costes de las acciones** elegidos cuidadosamente al realizar un análisis del coste actual de encendido y apagado de un sistema de calefacción. Esto nos permite calcular la política tomando una temperatura como estado y retornando una acción on o off, que debería ejecutarse en dicho estado para alcanzar la temperatura deseada con menor coste. Para definir la política óptima calcularemos unos valores para cada estado y los usaremos para escoger la mejor acción:

- $V(temp)$ : el valor de la **temperatura** es el coste que esperamos pagar para llegar desde esa **temperatura** a la meta (**temperatura deseada**) si tomamos la mejor decisión posible en cada estado.

Los valores de la temperaturas se calculan por ciclos siguiendo la Ecuación de Bellman:

$$V_{i+1}(temp) = \min(Coste + \sum_{t=16}^{25} prob_{on/off} * V_i(temp_t)) .$$

Estos valores inician en 0 y van sustituyéndose los nuevos (obtenidos con la ecuación) en el siguiente ciclo hasta que converjan. La acción óptima será la acción que ha llevado al último valor (el mínimo).

De esta forma, conseguimos implementar satisfactoriamente un MDP con su resolución en el código fuente que se puede encontrar en el módulo [main.py](#).

## DESCRIPCIÓN FORMAL DEL MODELO MDP

Para alcanzar el entendimiento total del problema debemos, en primer lugar, comprender su implicación teórica. Para ello, analizaremos todos los componentes de la función a implementar.

El problema presenta una calefacción la cual va variando entre las temperaturas posibles, es decir, las posibles temperaturas que este puede detectar se traducen en los **estados** del sistema. Como el enunciado indica, la temperatura estará entre los **16° y 25° en intervalos de 0,5°**.

El sistema debe decidir si **encenderse o apagarse** cada media hora, esta acción conlleva con cierta probabilidad un cambio de temperatura. Esto quiere decir que las posibles **acciones** que tiene el sistema de calefacción son esas dos. Por otro lado, dado un estado y una acción ejecutada habrá un conjunto de probabilidades de lo que ocurrirá posteriormente en la temperatura medida por el termostato (matrices de transición).

En resumen, partimos de un conjunto de estados (temperaturas), con un conjunto de acciones, las cuales implican un coste diferente para cada una de ellas y una matriz de probabilidades la cual se puede hallar a partir de las probabilidades mencionadas en el enunciado.

Nuestro **objetivo** es que dado un estado inicial cualquiera, obtener la acción más rentable para alcanzar la temperatura deseada. Por lo tanto, uno de nuestros estados se considerará el estado final y como dice el enunciado, una vez alcanzado dicho estado el sistema no debe tomar ninguna decisión.

Con el conjunto de datos y el objetivo a alcanzar que tenemos definidos concluimos que debemos implementar un Modelo de Markov pues representa problemas de política óptima en los que el estado actual influye en el estado inmediatamente siguiente.

- El conjunto de **estados** S serán las temperatura desde los 16° hasta los 25° con un salto de 0,5° es decir:  
 $S = [16, 16.5, 17, 17.5, 18, 18.5, 19, 19.5, 20, 20.5, 21, 21.5, 22, 22.5, 23, 23.5, 24, 24.5, 25]$
- El conjunto de **acciones** A será:  
 $A = [ON, OFF]$
- El estado final, como ya hemos mencionado anteriormente, está definido en 22°.

La matriz de transición asociada a cada una de las acciones no se incluirá en este apartado por simplicidad en el documento. Sin embargo, se incluye en el apartado de [FASES DEL PROYECTO](#) donde se vuelve a hacer referencia a ella.

Una vez definidas todas las partes claves del MDP podemos continuar con el desarrollo del proyecto.

## ANÁLISIS DETALLADO DEL MODELO DE COSTES

Puesto que el enunciado no hace mención ninguna acerca de los costes que conllevan cada una de las posibles acciones del sistema, dedicaremos un apartado del documento a desarrollar y justificar los valores atribuidos a estos costes. No obstante, en la implementación de la funcionalidad se han valorado las posibles variaciones que pueden tomar estos valores **permitiendo su modificación de manera muy sencilla**.

Para obtener los valores de los costes de las acciones del sistema de la mejor manera posible realizaremos una investigación acerca de los costes energéticos, en consecuencia se definirá un valor aproximado que sea coherente con la situación energética actual. Por otro lado, se hará una segunda investigación acerca del consumo de los sistemas de calefacción que se encuentran en el mercado, se tomarán varios ejemplos y se hará una media del consumo para obtener un valor aproximado del posible consumo de nuestro sistema.

Para obtener el valor de los costes de energía actuales buscaremos un valor actualizado, preferiblemente de un mes cercano al actual. Según la [Organización de Consumidores y Usuarios \(OCU\)](#) el precio de la luz en el mes de marzo de este año 2023 asciende a **44€/MWh** ( $44€/10^4 KWh = 0.0044€/KWh$ ). Fijamos este dato como el precio de la luz a utilizar para hallar el resto de nuestros costes.

Una vez definido el primer valor, continuaremos con nuestra labor de investigación. Esta vez para obtener el consumo medio de un sistema calefactor que se encuentre en el mercado actual. Este segundo caso supone muchas más variables a tener en cuenta como pueden ser la gama del sistema de calefacción o si es de gas o eléctrica. Según [calderasgascalfaccionzaragoza.com](#) y [energygo.es](#) el consumo medio de una caldera de gas asciende a **24 KWh**, por otro lado el consumo de una caldera eléctrica aumenta el valor con respecto al anterior, ofreciendo un consumo de **37,63 KWh**. Para definir un valor lo más cercano a ambos, obtendremos la media y ese será el consumo de nuestro supuesto sistema:

$$\text{Consumo medio} = \frac{24 + 37,63}{2} KWh = 30,815 KWh$$

Tras fijar estos 2 valores es momento de hallar el coste que supondría encender nuestro sistema. Si son 0.0044€ el consumo de 1KW durante una hora, podemos obtener el coste de encender la calefacción (acción ON) durante ese tiempo:

$$30,815 KWh * \frac{0.0044€}{1 KWh} = 0,135586€$$

Por lo tanto, como nuestro sistema se queda encendido durante media hora, el coste de la acción ON será **coste\_ON = 0,067793€**

---

Para hallar el coste de apagar el sistema utilizamos la información hallada en [xataka.com](http://xataka.com) la cual menciona que los aparatos conectados a la red eléctrica aunque estén apagados consumen alrededor de un 9% de lo que consumen si estuviesen encendidos (concepto del *consumo fantasma*). Es por ello que el coste de apagar la calefacción lo fijaremos como el 9% del coste de encenderla:

$$0,067793\text{€} * 0,09 = 0,00610137\text{€}$$

Entonces, el coste de la acción *OFF* será **coste\_OFF = 0,00610137€**.

## POLÍTICA ÓPTIMA PARA LOS COSTES ELEGIDOS

Para la obtención de la acción más óptima desde cualquier estado posible hasta la temperatura deseada, ejecutamos nuestra implementación con los siguientes datos:

```
#----- VALORES MODIFICABLES -----  
  
# Temperatura que se desea alcanzar (estado final)  
TEMPERATURA_DESEADA = 22  
# Coste de la acción "encender calefacción"  
COSTE_ON = 6.7793  
# Coste de la acción "apagar calefacción"  
COSTE_OFF = 0.610137  
# Ciclos que se desean para calcular la política óptima  
# En el caso de indicar None, se harán todos los ciclos necesarios  
# hasta converger  
N_CICLOS = None  
# Ruta al fichero csv que guarda las matrices con las probabilidades  
CSV_PROBS_ON = "./PROB_ON.csv"  
CSV_PROBS_OFF = "./PROB_OFF.csv"
```

Img1. Captura de los valores establecidos para obtener la política óptima con los costes escogidos

Nótese que hemos normalizado los costes, con tal de simplificar los cálculos. Para ello hemos multiplicado por 100 el valor de ambos costes. De esta manera la diferencia entre ellos se mantiene proporcional pero los números con los que se opera son más legibles, lo que ayudará bastante en las comprobaciones posteriores.

El resultado obtenido para un número de ciclos indefinido ( $N\_CICLOS = None$ ), es decir, hasta que los valores convergen, es el siguiente:

```
Temperatura deseada: 22  
Estado: 16.0 Política óptima: OFF  
Estado: 16.5 Política óptima: ON  
Estado: 17.0 Política óptima: ON  
Estado: 17.5 Política óptima: ON  
Estado: 18.0 Política óptima: ON  
Estado: 18.5 Política óptima: ON  
Estado: 19.0 Política óptima: ON  
Estado: 19.5 Política óptima: ON  
Estado: 20.0 Política óptima: ON  
Estado: 20.5 Política óptima: ON  
Estado: 21.0 Política óptima: ON  
Estado: 21.5 Política óptima: ON  
Estado: 22.0 Política óptima: TEMPERATURA DESEADA  
Estado: 22.5 Política óptima: OFF  
Estado: 23.0 Política óptima: OFF  
Estado: 23.5 Política óptima: OFF  
Estado: 24.0 Política óptima: OFF  
Estado: 24.5 Política óptima: OFF  
Estado: 25.0 Política óptima: OFF
```

Img2. Captura de la política óptima obtenida con los costes escogidos y sin ciclos establecidos



Por otro lado, cabe añadir que para comprobar la veracidad de los resultados obtenidos hemos realizado los cálculos “manualmente” en la hoja de cálculo *Comprobación\_cálculos.xlsx*. En esta hoja hemos realizado 24 ciclos con los costes ya mencionados y hemos comparado los valores obtenidos con los calculados por nuestro programa, imprimiendo éstos al final de cada ciclo.

Incluimos a continuación una captura de pantalla de 16 de los 24 cálculos realizados por nosotras, sin embargo, el fichero se incluye en la entrega para su mayor visibilidad:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	CICLO:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
2	16	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
3	16.5	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
4	17	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
5	17.5	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
6	18	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
7	18.5	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
8	19	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
9	19.5	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
10	20	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
11	20.5	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
12	21	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
13	21.5	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
14	22	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
15	22.5	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
16	23	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
17	23.5	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
18	24	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
19	24.5	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
20	25	0	0.610137	1.220274	1.830411	2.440548	3.050685	3.660822	4.270959	4.881096	5.491233	6.10137	6.711507	7.321644	7.931781	8.541918	9.152055	9.762192	10.3723
21																			
22																			
23																			
24		6,7793	<- Coste ON																
25		0,61013	<- Coste OFF																

Img3. Captura de los valores de los estados obtenidos de forma manual para su posterior comprobación

Para comprobar que nuestro código realiza de forma correcta los cálculos, hemos impreso el valor de los estados que calcula en cada ciclo:

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

['VALORES ESTADOS: ', 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137, 0.610137]

['VALORES ESTADOS: ', 1.2202740000000003, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999, 1.2202739999999999]

['VALORES ESTADOS: ', 1.8304110000000002, 1.8304110000000002, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998, 1.8304109999999998]

['VALORES ESTADOS: ', 2.440548, 2.440548, 2.440548, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997, 2.4405479999999997]

['VALORES ESTADOS: ', 3.0506850000000005, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996, 3.0506849999999996]

['VALORES ESTADOS: ', 3.6608220000000005, 3.660822, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996, 3.6608219999999996]

['VALORES ESTADOS: ', 4.270959, 4.270959, 4.270959, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995, 4.2709589999999995]

['VALORES ESTADOS: ', 4.881096, 4.881096, 4.881096, 4.881096, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999, 4.8810959999999999]

['VALORES ESTADOS: ', 5.4912330000000001, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999, 5.4912329999999999]
```

Img4. Captura de los valores de los estados obtenidos por el código implementado para su posterior comprobación

Como se puede observar, los valores coinciden perfectamente, lo que nos asegura que por más extraño que parezca la política óptima, ésta es la que corresponde. Por lo tanto, la política óptima obtenida para esos costes y en 24 ciclos es:

```
[ 'VALORES ESTADOS: ', 14.64328799940562, 14.643287997871264, 14.643287983488603, 14.643287861901715, 14.643286864257327, 14.643278840464449, 14.64321425315493, 14.642661728576897, 14.637618810057694, 14.590645115599198, 14.162922860195957, 10.414723139116731, 0, 1.01686551577398, 2.0336202484201125, 3.0498522126319787, 4.06318208749628, 5.058091558417111, 5.9279390037927975]

Temperatura deseada: 22
Estado: 16.0 Política óptima: OFF
Estado: 16.5 Política óptima: OFF
Estado: 17.0 Política óptima: OFF
Estado: 17.5 Política óptima: OFF
Estado: 18.0 Política óptima: OFF
Estado: 18.5 Política óptima: OFF
Estado: 19.0 Política óptima: OFF
Estado: 19.5 Política óptima: OFF
Estado: 20.0 Política óptima: OFF
Estado: 20.5 Política óptima: OFF
Estado: 21.0 Política óptima: OFF
Estado: 21.5 Política óptima: ON
Estado: 22.0 Política óptima: TEMPERATURA DESEADA
Estado: 22.5 Política óptima: OFF
Estado: 23.0 Política óptima: OFF
Estado: 23.5 Política óptima: OFF
Estado: 24.0 Política óptima: OFF
Estado: 24.5 Política óptima: OFF
Estado: 25.0 Política óptima: OFF
```

Img5. Captura de los valores finales obtenidos y la política óptima con 24 ciclos

z
24
14,643288
14,643288
14,6432879
14,6432878
14,6432868
14,6432788
14,6432142
14,6426617
14,6376188
14,5906451
14,1629228
10,4147231
0
1,01686551
2,03362024
3,04985221
4,06318208
5,05809155
5,92793900

Se puede comprobar que el valor de los estados en el último ciclo es exactamente el mismo que el calculado en la hoja de cálculo (Img6).

Img6.

## FASES DEL PROYECTO

### Diseño

Con el fin de desarrollar un código fuente que permita obtener la política óptima para la calefacción (acción ON o acción OFF) dada la temperatura que se desea alcanzar y desde cualquiera de los estados iniciales posibles (del 16 a 25 grados), hemos creado en primer lugar un repositorio compartido en [github](#) y una carpeta compartida en [drive](#) para agilizar el trabajo y planificación en equipo.

Primeramente, realizamos las matrices de transición siguiendo los datos proporcionados por el enunciado en una hoja de cálculo (una para cada acción). Éstas indican la probabilidad de transitar a otro estado según la acción elegida. Una vez terminadas, las descargamos en formato .csv para que puedan ser leídas con facilidad usando el lenguaje de programación *Python* (estos ficheros se encuentran en la misma carpeta que el código fuente *main.py* con el nombre *PROB\_ON.csv* y *PROB\_OFF.csv*).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	ON	16	16.5	17	17.5	18	18.5	19	19.5	20	20.5	21	21.5	22	22.5	23	23.5	24	24.5	25
2		16	0.30	0.50	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3		16.5	0.10	0.20	0.50	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4		17	0.00	0.10	0.20	0.50	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5		17.5	0.00	0.00	0.10	0.20	0.50	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6		18	0.00	0.00	0.00	0.10	0.20	0.50	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7		18.5	0.00	0.00	0.00	0.00	0.10	0.20	0.50	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8		19	0.00	0.00	0.00	0.00	0.00	0.10	0.20	0.50	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9		19.5	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20	0.50	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10		20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20	0.50	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11		20.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20	0.50	0.20	0.00	0.00	0.00	0.00	0.00	0.00
12		21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20	0.50	0.20	0.00	0.00	0.00	0.00	0.00
13		21.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20	0.50	0.20	0.00	0.00	0.00	0.00
14		22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20	0.50	0.20	0.00	0.00	0.00
15		22.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20	0.50	0.20	0.00	0.00
16		23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20	0.50	0.20	0.00
17		23.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20	0.50	0.20
18		24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20	0.50
19		24.5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.20
20		25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.10	0.90

Img7. Hoja de cálculo con la matriz de transición de la acción ON

```
PROB_OFF.csv
1 OFF,16,16.5,17,17.5,18,18.5,19,19.5,20,20.5,21,21.5,22,22.5,23,23.5,24,24.5,25
2 16,0.90,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
3 16.5,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
4 17,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
5 17.5,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
6 18,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
7 18.5,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
8 19,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
9 19.5,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
10 20,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
11 20.5,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
12 21,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
13 21.5,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00,0.00
14 22,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00,0.00
15 22.5,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00,0.00
16 23,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00,0.00
17 23.5,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00,0.00
18 24,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00,0.00
19 24.5,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.20,0.10,0.00
20 25,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.70,0.30
```

Img8. Fichero .csv que se encuentra junto al código de la acción OFF

En segundo lugar, realizamos un MDP conceptual (ver capítulo [DESCRIPCIÓN FORMAL DEL MODELO MDP](#)) y empezamos la implementación en código ([main.py](#)). En este paso, empezamos leyendo las matrices de transición haciendo uso de la [librería csv](#) y almacenando los valores en una lista de listas (equivalente a una matriz). De esta forma obtenemos dos matrices, la de la acción ON (*matriz\_ON*) y la de la acción OFF (*matriz\_OFF*). Una vez obtenidos los valores, empezamos a implementar los cálculos de los valores siguiendo la [Ecuación de Bellman](#).

Con el propósito de almacenar los nuevos valores de los estados para poder usarlos en los cálculos del siguiente ciclo, realizamos una función recursiva que iba recibiendo como parámetro la *lista\_convergencia*, la cual se iba actualizando con los nuevos valores al terminar un ciclo y la *lista\_politica\_optima* que almacenaba la acción que llevaba al nuevo valor (el mínimo entre las dos posibilidades, ya que buscamos la política óptima). Este diseño nos llevó a ver que cuando unos valores tardaban mucho en converger se excedía la recursividad, saltado el error *Maximum Recursion Exceeded*. Por esta razón, cambiamos nuestro diseño a un bucle **while**.

Por lo tanto, nuestro diseño final siguiente estructura:

- VALORES MODIFICABLES: Al principio del código nos encontramos con las constantes que pueden ser modificadas para hacer el código lo más general posible. Éstas son la temperatura deseada, el coste de encendido, el coste de apagado, el número de ciclos para realizar los cálculos y la ruta de los archivos .csv con las matrices de transición.
- OBTENCIÓN DEL RESTO DE VARIABLES: Incluye una función generada por nosotras que permite leer una tabla en .csv y convertirla en una matriz de *Python*, la cual usamos en la misma sección para obtener las dos matrices. También incluye la obtención de la lista de convergencia inicial (valores de los estados) inicializada a 0.
- COMPROBACIÓN DE VARIABLES: Comprobamos que las variables que usaremos para calcular la política óptima son correctas. Por ejemplo, que la temperatura deseada establecida por el usuario esté dentro de las posibles temperaturas valoradas en este proyecto.
- CÁLCULO DE LA POLÍTICA ÓPTIMA: Nos recorremos cada estado (temperatura inicial) de la matriz para obtener sus probabilidades de transición y calcular su siguiente valor, haciendo uso del valor anterior de todos los estados mediante una lista que los va almacenando. También se almacena la acción óptima que lo lleva a tal valor. Este bucle no para de calcular hasta que converjan los valores o hasta llegar a los ciclos indicados por el usuario.
- RESULTADOS: Mostramos los resultados por pantalla siguiendo el formato  
"Temperatura deseada: 22  
Estado: 16.0 Política óptima: OFF  
Estado: 16.5 Política óptima: OFF  
..."

## Implementación

Se desarrolla un código fuente en el módulo `main.py` escrito en el lenguaje de programación *Python*, el cual presenta el código que permite obtener la política óptima para la calefacción (acción ON o acción OFF) dada la temperatura que se desea alcanzar. Es decir, se presentará la acción más factible desde cada uno de los estados posibles en los que te puedes encontrar inicialmente (del 16 a 25 grados).

Para ello, debe darse en primer lugar unos valores a las constantes temperatura deseada (`TEMPERATURA_DESEADA`), coste de encendido (`COSTE_ON`), coste de apagado (`COSTE_OFF`) y el número de ciclos (`N_CICLOS`) que se desea alcanzar para obtener el valor de los estados con la Ecuación de Bellman (puede establecerse a *None* para realizar todos los ciclos necesarios hasta converger). Si se desea, también puede darse una ruta distinta de los archivos `.csv` que guardan las tablas de probabilidades de las dos acciones cambiando las constantes `CSV_PROBS_ON` y `CSV_PROBS_OFF`. Las tablas usadas por nosotras se encuentran en la misma carpeta que el código fuente bajo el nombre de `PROB_ON.csv` y `PROB_OFF.csv`.

En segundo lugar, se implementa la función `obtener_matrices_probs(csv_path)`, la cual lee un `.csv` pasado como argumento y cada línea la guarda en una lista que se incluirá dentro de una matriz. Después se hace uso de esa función para obtener `matriz_ON` y `matriz_OFF`, las cuales guardan las matrices de transición de la acción on y off respectivamente. Seguidamente, se inicializa `lista_convergencia` a 0, ya que es la lista que guarda los valores de los estados, y en el ciclo 0 los valores son todos cero.

Después, se validan las variables que serán usadas en los cálculos para encontrar la política óptima: la temperatura deseada debe ser una de las posibles, el coste de apagado y el de encendido debe ser un real, y el número de ciclos debe ser *None* o un entero mayor que 0.

Más adelante, nos encontramos con el cálculo de la política óptima. Para posibilitar su buen funcionamiento, hay que realizar un bucle ***while*** que seguirá calculando los valores de los estados ciclo tras ciclo hasta que éstos converjan o, en caso de ser especificado, se alcance el ciclo indicado.

En el interior del ***while*** nos encontramos con un bucle ***for*** que se recorre la matriz línea a línea. La matriz contiene en su primera línea los estados y no las probabilidades, es por ello que realizamos un ***if*** que permite saltarnos los cálculos en esta primera iteración (*if estado != 0*). Si no es la primera línea de la matriz, establecemos el valor del estado respecto a la acción on (`v_on`) y el valor respecto a la acción off (`v_off`) a 0 y obtenemos las probabilidades del estado en el que nos encontramos extrayendo la línea de ambas matrices. Una vez obtenidas, realizamos dos ***for's*** más que permiten recorrerse dichas listas con las probabilidades de transición de dicho estado (`lista_probs_ON` y `lista_probs_OFF`) y realizar los cálculos. Al recorrernos la lista, si la probabilidad es distinta de 0 la multiplicamos por el valor anterior del estado al que se refiere dicha probabilidad y



se lo sumamos al valor del estado que estamos calculando. Al terminar el sumatorio, sumamos el coste de la acción en la cual nos encontremos.

Una vez obtenido el valor del estado, si no se han especificado ciclos, lo redondeamos a 10 decimales para asegurar que éstos converjan y acabe la ejecución del código. Posteriormente, si no nos encontramos en el estado final (temperatura deseada) comprobamos qué valor respecto a las dos acciones es menor y lo añadimos a la lista que contiene los valores actualizados (*lista\_convergencia\_temp*). También, añadimos la acción que nos lleva a dicho valor en la *lista\_politica\_optima*. En caso de encontrarnos en el estado final, añadimos 0 a la lista que contiene los valores, ya que el valor de un estado final en MDP es 0. Además, añadimos “TEMPERATURA DESEADA” en *lista\_politica\_optima* para diferenciarlo del resto.

Al terminar todos los cálculos y tener la nueva lista con los valores actualizados, le restamos 1 al número de ciclos indicado en caso de existir (para poder parar cuando se nos indica) y comprobamos si la lista de valores obtenida (*lista\_convergencia\_temp*) es igual a la inicial (*lista\_convergencia*) para ver si han convergido los valores. En caso de converger, establecemos la variable *converge* a *True* lo que fuerza a salir del *while*. En caso contrario, establecemos la nueva lista de convergencia inicial como la obtenida en este ciclo, es decir, *lista\_convergencia = lista\_convergencia\_temp*.

Finalmente, saldremos del *while* con la *lista\_politica\_optima* actualizada y correcta, por lo que podemos proceder a imprimir los resultados. Para ello, mostramos primero la temperatura deseada y luego la política óptima según cada estado.

## Pruebas

A continuación se presenta la batería de pruebas para comprobar la funcionalidad del código.

Las pruebas que se presentan consistirán en la variación de los parámetros de, *COSTE\_ON* y *COSTE\_OFF*. La variable *N\_CICLOS* será definida en todas las pruebas como *None*, de esta manera nos aseguramos que todos los valores obtenidos en las pruebas convergen.

Cabe mencionar que las pruebas relacionadas con la variación de *N\_CICLOS* ya han sido mostradas en el apartado de [POLÍTICA ÓPTIMA PARA LOS COSTES ELEGIDO](#), donde también se comprueba que los cálculos de nuestro código son correctos.

En primer lugar, definiremos el mismo valor para las dos acciones posibles de esta manera el resultado esperado será ON para aquellos estados que estén por debajo del ideal mientras que para aquellos que se encuentren por encima del estado ideal la acción recomendada será OFF:

```
COSTE_ON = 1
# Coste de la acción "apagar calefacción"
COSTE_OFF = 1
# Ciclos que se desean para calcular la política
# En el caso de indicar None, se harán todos los
# hasta conseguir...

main x
Temperatura deseada: 22
Estado: 16.0 Política óptima: ON
Estado: 16.5 Política óptima: ON
Estado: 17.0 Política óptima: ON
Estado: 17.5 Política óptima: ON
Estado: 18.0 Política óptima: ON
Estado: 18.5 Política óptima: ON
Estado: 19.0 Política óptima: ON
Estado: 19.5 Política óptima: ON
Estado: 20.0 Política óptima: ON
Estado: 20.5 Política óptima: ON
Estado: 21.0 Política óptima: ON
Estado: 21.5 Política óptima: ON
Estado: 22.0 Política óptima: TEMPERATURA DESEADA
Estado: 22.5 Política óptima: OFF
Estado: 23.0 Política óptima: OFF
Estado: 23.5 Política óptima: OFF
Estado: 24.0 Política óptima: OFF
Estado: 24.5 Política óptima: OFF
Estado: 25.0 Política óptima: OFF
```

Img9. El resultado esperado coincide con el obtenido.

La segunda prueba consistirá en poner uno de los costes a 0, en nuestro caso será el de la acción de apagar. Esto supondrá que la política óptima en todos los estados sea la de apagar:

```
COSTE_ON = 1
# Coste de la acción "apagar calefacción"
COSTE_OFF = 0
# Ciclos que se desean para calcular la política óptima

main x
Temperatura deseada: 22
Estado: 16.0 Política óptima: OFF
Estado: 16.5 Política óptima: OFF
Estado: 17.0 Política óptima: OFF
Estado: 17.5 Política óptima: OFF
Estado: 18.0 Política óptima: OFF
Estado: 18.5 Política óptima: OFF
Estado: 19.0 Política óptima: OFF
Estado: 19.5 Política óptima: OFF
Estado: 20.0 Política óptima: OFF
Estado: 20.5 Política óptima: OFF
Estado: 21.0 Política óptima: OFF
Estado: 21.5 Política óptima: OFF
Estado: 22.0 Política óptima: TEMPERATURA DESEADA
Estado: 22.5 Política óptima: OFF
Estado: 23.0 Política óptima: OFF
Estado: 23.5 Política óptima: OFF
Estado: 24.0 Política óptima: OFF
Estado: 24.5 Política óptima: OFF
Estado: 25.0 Política óptima: OFF
```

Img10. El resultado esperado coincide con el obtenido.

La siguiente prueba consiste en definir costes con mucha diferencia. Comprobaremos que el código puede manejar números muy grandes sin problemas. El resultado esperado será que devuelva la política óptima de cada uno de los estados:

```
COSTE_ON = 10000
# Coste de la acción "apagar calefacción"
COSTE_OFF = 1

le ((not converge) and ((not...  if N_CICLOS

main x
Estado: 16.0 Política óptima: OFF
Estado: 16.5 Política óptima: OFF
Estado: 17.0 Política óptima: OFF
Estado: 17.5 Política óptima: OFF
Estado: 18.0 Política óptima: ON
Estado: 18.5 Política óptima: ON
Estado: 19.0 Política óptima: ON
Estado: 19.5 Política óptima: ON
Estado: 20.0 Política óptima: ON
Estado: 20.5 Política óptima: ON
Estado: 21.0 Política óptima: ON
Estado: 21.5 Política óptima: ON
Estado: 22.0 Política óptima: TEMPERATURA DESEADA
Estado: 22.5 Política óptima: OFF
Estado: 23.0 Política óptima: OFF
Estado: 23.5 Política óptima: OFF
Estado: 24.0 Política óptima: OFF
Estado: 24.5 Política óptima: OFF
Estado: 25.0 Política óptima: OFF
```

Img11. El resultado esperado coincide con el obtenido.

Es momento de ejecutar la prueba complementaria a la realizada anteriormente. Probaremos que el código se maneja bien con valores muy pequeños. El resultado esperado será la devolución de la política óptima para cada uno de los estados:

```
COSTE_ON = 1
# Coste de la acción "apagar calefacción"
COSTE_OFF = 0.0001

main x
Temperatura deseada: 22
Estado: 16.0 Política óptima: OFF
Estado: 16.5 Política óptima: OFF
Estado: 17.0 Política óptima: OFF
Estado: 17.5 Política óptima: OFF
Estado: 18.0 Política óptima: ON
Estado: 18.5 Política óptima: ON
Estado: 19.0 Política óptima: ON
Estado: 19.5 Política óptima: ON
Estado: 20.0 Política óptima: ON
Estado: 20.5 Política óptima: ON
Estado: 21.0 Política óptima: ON
Estado: 21.5 Política óptima: ON
Estado: 22.0 Política óptima: TEMPERATURA DESEADA
Estado: 22.5 Política óptima: OFF
Estado: 23.0 Política óptima: OFF
Estado: 23.5 Política óptima: OFF
Estado: 24.0 Política óptima: OFF
Estado: 24.5 Política óptima: OFF
Estado: 25.0 Política óptima: OFF
```

Img12. El resultado esperado coincide con el obtenido.



## PRESUPUESTO

A continuación se hará una estimación económica acerca del presupuesto que hubiese sido necesario para el estudio, desarrollo e implementación de este proyecto si fuese un caso real.

Para realizar una estimación del presupuesto completo se deben tener en cuenta varios factores determinantes. A continuación se enumeran dichos factores, acompañados además de un desglose del presupuesto que se debería dedicar a éstos para completar el proyecto en el tiempo y forma requeridos.

En primer lugar, se debe tener en cuenta el tiempo dedicado al desarrollo del proyecto. Empezando por el tiempo invertido en la comprensión del problema y de su entorno, sumando además la investigación que esto implica. El tiempo que se dedicó a esta primera etapa fue de un total de **2 horas**. Para el desarrollo, implementación y depuración del código presentado han sido necesarios alrededor de **2 días** de trabajo lo que se puede traducir en 12 horas dedicadas al código. Además, como ya se ha mencionado en el apartado anterior, han sido propuestos varios modelos de implementación de la funcionalidad (cambio de función recursiva a un bucle while), para presentar la mejor implementación posible.

Por otro lado, se debe tener en cuenta el tiempo dedicado a las comprobaciones de la funcionalidad. Estas han sido un tanto tediosas puesto que hemos querido asegurar que los cálculos que se generaban por el código eran realmente los que se esperaban y la única manera de comprobar dicha funcionalidad era hacerlo sin código. Para la comprobación de la funcionalidad se han destinado **2 horas** más de trabajo.

Finalmente debemos incluir el tiempo dedicado a la documentación del proyecto por medio del presente documento. Puesto que se han tenido que incluir varios puntos un tanto extensos además de la grabación de un video explicatorio acerca de la funcionalidad del propio código, el tiempo dedicado ha sido un tanto abultado y mayor al que el equipo esperaba. El tiempo dedicado al desarrollo de este apartado ha sido de un día entero de trabajo, es decir, alrededor de **8 horas** de trabajo.

El tiempo total invertido en el desarrollo completo del proyecto ha sido de **24 horas** de trabajo por cada integrante del equipo.

Según [talent.com](https://www.talent.com) el sueldo medio de un graduado en Ingeniería Informática asciende a los 2808€ al mes. Suponiendo que el mes tiene 20 días hábiles, el sueldo por jornada laboral es de **140.4€**. Al tratarse de un proyecto al que hay que dedicarle lo equivalente a 3 jornadas laborales, el presupuesto dedicado al sueldo del equipo debe ser 140.4€ multiplicado por 3 días de trabajo. Esto supone un total de **421€ por participante**. En nuestro caso, el proyecto se llevó a cabo en un equipo compuesto por 2 personas, lo que supone que el presupuesto asciende a **842€**.

---

En segundo lugar, debemos tener en cuenta el coste de los recursos necesarios para el desarrollo del código, así como el coste de otras herramientas que sean necesarias para cumplir con el resto de requisitos del proyecto. El equipo ha supuesto que las herramientas básicas tales como acceso a internet, luz o equipos donde desarrollar el proyecto serán proporcionadas por la propia empresa y por ende no supondrán ningún desembolso adicional. En caso de que esto no fuese así, se tendría que tener en cuenta destinar un presupuesto a cumplir con los 3 puntos mencionados anteriormente.

En conclusión, el presupuesto que debería destinarse al desarrollo del proyecto oscila los 850€. Se recomienda mantener un colchón de 150€ en caso de ser necesario destinar parte de esta cuantía en imprevistos que haya que solucionar, cerrando el **presupuesto total del proyecto en 1000€**.

---

## CONCLUSIONES

Una vez expuestos todos los puntos que consideramos que han sido determinantes para el correcto desarrollo de la funcionalidad requerida, nos gustaría compartir las conclusiones alcanzadas tras la finalización del proyecto.

Desde nuestro punto de vista, el proyecto ha superado con creces los objetivos a cumplir. El hecho de tener tanto tiempo para solucionar el problema propuesto ha permitido que la implementación de éste se haya podido ir mejorando con el paso del tiempo, alcanzando una versión del código que consideramos inmejorable con los datos proporcionados en el enunciado. Este aspecto sumado, además, a los conocimientos teórico-prácticos obtenidos a lo largo de la asignatura han facilitado en gran medida el desarrollo del proyecto.

Por otro lado, la aplicación de estos conocimientos teóricos adquiridos con anterioridad ha sido muy provechosa puesto que nos ha ayudado a afianzarlos completamente y entender por medio de una aplicación práctica su utilidad.

Nos gustaría resaltar también algunos problemas que nos hemos encontrado durante el desarrollo de esta práctica. El primero y principal fue el problema ya mencionado en el apartado de fases del proyecto. El hecho de desarrollar un código al completo y no obtener los resultados esperados debido a un problema con la gestión de recursos de *Python* hizo que el principio de la práctica se hiciera complicado. Una vez propuesta la segunda versión del código en la que no tuvimos ningún problema con los recursos, la práctica siguió un desarrollo constante. Aunque nos enfrentamos a otros problemas como el entendimiento del enunciado propuesto o el desarrollo de algunos puntos de la memoria, éstos fueron mínimos y solucionados en muy poco tiempo.

Como conclusión nos gustaría resaltar que hemos disfrutado mucho durante el desarrollo de esta práctica y de la asignatura en general. Por un lado, el caso que se planteaba estaba relacionado con la vida cotidiana lo que hizo que el entendimiento del entorno del problema se hiciera más sencillo de lo que solemos recibir en otras prácticas. Por otro lado, la propuesta del enunciado inmediatamente después de la realización del examen parcial cuyos temas a evaluar estaban directamente relacionados con los conocimientos necesarios para el desarrollo de la práctica también ha facilitado la realización de este proyecto.

Cabe resaltar nuestra completa conformidad con la realización de una única práctica final. A lo largo de este cuatrimestre hemos tenido que desarrollar una infinidad de prácticas que se solapaban unas entre otras y que ocupaban gran parte de nuestro tiempo de estudio. El hecho de desarrollar una única práctica al final nos parece una muy buena alternativa, puesto que cumple con la aplicación de los conocimientos mientras que dejas más tiempo para otras labores.

---

Finalmente, agradecer la buena organización de la asignatura y vuestra labor como docentes.

---

## BIBLIOGRAFÍA

Precio luz mes de marzo 2023: Organización de Consumidores y Usuarios (OCU). Precio de la luz. <https://www.ocu.org/vivienda-y-energia/gas-luz/informe/precio-luz>

Consumos según el tipo de caldera:

- Caldera de gas:  
<https://calderasgascalfaccionzaragoza.com/cuanto-consume-caldera-de-gas/#:~:text=Una%20caldera%20de%2024%20kW%20utilizar%C3%A1%2024%20kW%20de%20energ%C3%ADa,m%C3%A1s%20gas%20del%20que%20deber%C3%ADa.>
- Caldera eléctrica:  
<https://blog.energygo.es/caldera-electrica-o-caldera-de-gas-cual-consume-mas/#:~:text=Podemos%20coger%20como%20referencia%20unos,todos%20los%20d%C3%ADas%20del%20a%C3%B1o.>

Consumo fantasma:

<https://www.xataka.com/energia/consumo-fantasma-mitos-verdades-y-soluciones>

Salario ingeniero informático:

<https://es.talent.com/salary?job=ingeniero+inform%C3%A1tico#:~:text=El%20salario%20ingeniero%20inform%C3%A1tico%20promedio,hasta%20%E2%82%AC%2035.000%20al%20a%C3%B1o.>