Escala de gravamen del Impuesto a la renta

Base imponible	Cuota íntegra	Tipo
4.410.000	1.165.978	38,86%
4.830.000	1.329.190	41,02%
5.250.000	1.501.474	43,18%
5.670.000	1.682.830	

La cuota íntegra del Impuesto sobre la Renta se determina aplicando una fórmula basada en la interpolación lineal. Un contribuyente tiene una base imponible de 5 millones. Para calcular lo que tiene que pagar a Hacienda efectúa las siguientes operaciones, consultando la escala de gravamen anterior:

Base	5.000.000		Cuota
Hasta	4.830.000		1.329.190
Resto....	170.000	al 41,02%	69.734
		SUMA	1.398.924

El tipo marginal del 41,02% que aparece en la escala de gravamen es precisamente el cociente de las diferencias entre las cuotas íntegras y las bases imponibles más próximas en la escala a los 5 millones.

$$\frac{1.501.474 - 1.329.190}{5.250.000 - 4.830.000} = 0.4102$$

La fórmula aplicada es, en definitiva,

$$\text{Cuota} = 1.329.190 + 0{,}4102(\text{Base} - 4.830.000)$$

Para las bases comprendidas en el intervalo [4.830.000,5.250.000].

En particular, para una base imponible de 5.250.000 es indiferente aplicar la fórmula anterior o tomar directamente el valor de la tabla. En términos matemáticos esto equivale a decir que la Cuota es una función continua de la Base imponible.
El Impuesto sobre la Renta es progresivo, es decir, que el tipo de la imposición aumenta con la base imponible, como se comprueba observando la escala de gravamen. Así, el tipo medio correspondiente a 4.830.000 es el 27,52% y el de 5.250.000 es el 28,60%.

El contribuyente se siente perjudicado por el hecho de que al Resto de su Base imponible (170.000) se le aplica el mismo tipo marginal (41,02%) que, a otro contribuyente con una Base de 5.250.000, alegando que debe aplicársele el correspondiente a la base más próxima en la escala (4.830.000) que es del 38,86. Hacienda, por su parte, rechaza estos argumentos y efectúa la liquidación según sus normas. El sujeto del impuesto interpone recurso (tutela) ante el Tribunal competente, que considera en parte sus alegaciones. El fallo establece que en todo caso se debería aplicar un tipo marginal intermedio.

Como experto en temas fiscales debes elaborar un informe para que Hacienda conozca las diferencias entre el actual sistema impositivo y los posibles métodos de determinar la imposición correspondiente a la base de 5 millones por interpolación de segundo y tercer grado en la escala de gravamen.
¿En cada grado debe añadirse la base más próxima a 5 millones?

Realizado por:

Thomás Rivera

Juan Mejía

Nicolás Puerto

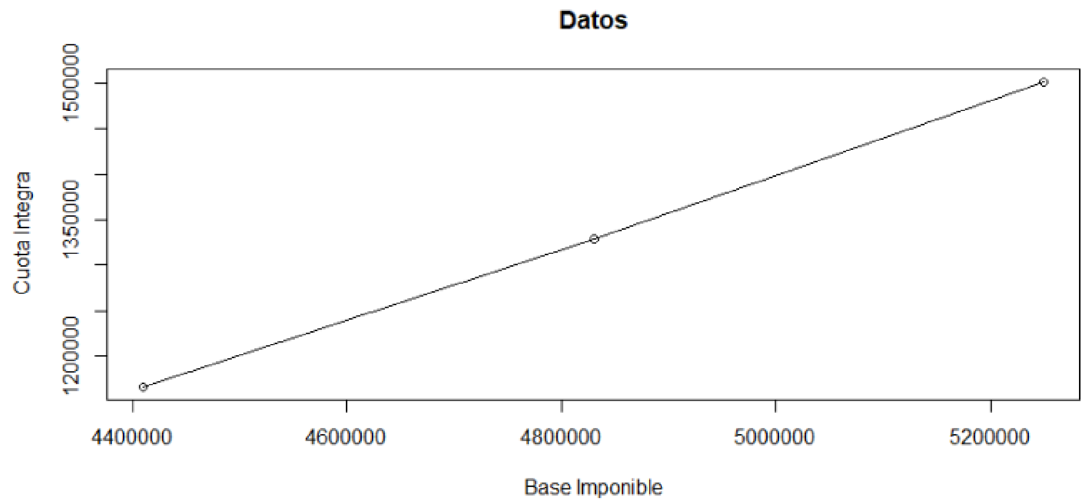
Para resolver el problema anteriormente planteado se utilizó:

La interpolación polinomial de Lagrange en segundo grado

Código en R

```r
#Lagrange
library(rSymPy)
library(polynom)

x <- c(4410000,4830000,5250000)
y <- c(1165978,1329190,1501477)

plot(x ,y , type = "o", main="Datos",xlab = "Base Imponible", ylab = "Cuota Integra")

lagrange.poly <- function(x, y) {

  l <- list()
  k <- 1

  for (i in x) {
    num <- 1
    denom <- 1

    p <- x[! x %in% i]

    for (j in p) {
      num <- paste(num, "*", "(", 'x', " - ", as.character(j), ")", sep = "", collapse = "")
      denom <- paste(denom, "*", "(", as.character(i)," - ", as.character(j), ")", sep = "", collapse = "")
    }

    l[k] <- paste("(", num, ")", "/", "(", denom, ")", sep = "", collapse = "")
    k <- k + 1
  }

  eq <- 0

  for (i in 1:length(y)) {
    eq <- paste(eq, '+', as.character(y[i]), "*", l[[i]], sep = "", collapse = "")
  }

  x <- Var('x')

  return(sympy(paste("simplify(", eq, ")")))
}

lagrange.poly(x, y)
poly.calc(x, y)

f <- function(x) {
  return(2.572279e-08*x^2 + 0.1509214*x + 155.125)
}
plot(f, type = "l", main="Comportamiento polinomio de Lagrage")
```

Se realizo el código correspondiente (la función "*lagrange.poly()*"), además fue encontrada la función "*poly.calc()*" de la librería "*polynom*" la cual nos permitió desarrollar el calculo de manera más sencilla. También se presenta el siguiente grafico con el fin de enseñar los datos del ejercicio:



Se puede observar un comportamiento directamente proporcional, es decir, al aumentar la base imponible incrementa la cuota íntegra.

Resultados

Se llegó al polinomio

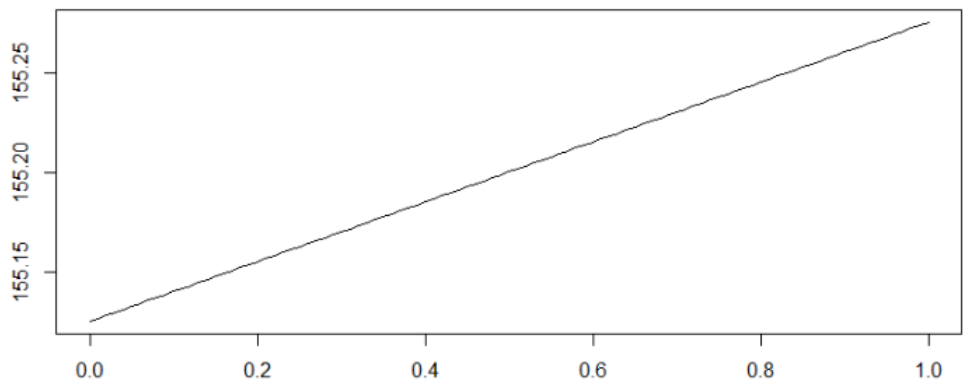
```
> lagrange.poly(x, y)
[1] "1241/8 + 21129*x/140000 + 121*x**2/4704000000"
>
> poly.calc(x, y)
155.125 + 0.1509214*x + 2.572279e-08*x^2
```

"poly.calc" llega a una respuesta simplificada, comparando con la herramienta online "*Dcode*" comprobamos la solución

Source : https://www.dcode.fr/lagrange-interpolating-polynomial

$$f(x) = \left(\frac{121(x-4830000)}{4704000000} + \frac{1943}{5000}\right)(x - 4410000) + 1165978 = \frac{121x^2}{4704000000} + \frac{21129x}{140000} + \frac{1241}{8}$$

Finalmente se muestra el comportamiento del polinomio encontrado



La interpolación polinomial de Lagrange en tercer grado

Código en R

```
#Lagrange
library(rSymPy)
library(polynom)

x <- c(4410000,4830000,5250000,5670000)
y <- c(1165978,1329190,1501477,1682830)

plot(x ,y , type = "o", main="Datos",xlab = "Base Imponible", ylab = "Cuota Integra")

lagrange.poly <- function(x, y) {

  l <- list()
  k <- 1

  for (i in x) {
    num <- 1
    denom <- 1

    p <- x[! x %in% i]

    for (j in p) {
      num <- paste(num, "*", "(", 'x', " - ", as.character(j), ")", sep = "", collapse = "")
      denom <- paste(denom, "*", "(", as.character(i)," - ", as.character(j), ")", sep = "", collapse = "")
    }

    l[k] <- paste("(", num, ")", "/", "(", denom, ")", sep = "", collapse = "")
    k <- k + 1
  }

  eq <- 0

  for (i in 1:length(y)) {
    eq <- paste(eq, '+', as.character(y[i]), "*", l[[i]], sep = "", collapse = "")
  }

  x <- Var('x')

  return(sympy(paste("simplify(", eq, ")")))
}

lagrange.poly(x, y)
poly.calc(x, y)


f <- function(x) {
  return(2.572279e-08*x^2 + 0.1509214*x + 155.125)
}
plot(f, type = "l", main="Comportamiento polinomio de Lagrage")
```

En caso se procede de la misma forma que en el anterior, la única diferencia radica en el aumento de puntos que permitió generar el grado del polinomio buscado.

Resultados

Se obtiene

```
> lagrange.poly(x, y)
[1] "38707/16 + 167449*x/1120000 + 6119*x**2/235200000000 - x**3/49392000000000000"
>
> poly.calc(x, y)
2419.187 + 0.149508*x + 2.601616e-08*x^2
```

Sin embargo la función "*poly.calc*" no logró calcular el polinomio de manera adecuada, contrastando lo obtenido con la herramienta se observa que solo el algoritmo llego a lo correcto.

Source : https://www.dcode.fr/lagrange-interpolating-polynomial

$$f(x) = \left(\left(\frac{5250000-x}{49392000000000000} + \frac{121}{4704000000} \right) (x - 4830000) + \frac{1943}{5000} \right) (x - 4410000) + 1165978$$
$$= -\frac{x^3}{49392000000000000} + \frac{6119x^2}{235200000000} + \frac{167449x}{1120000} + \frac{38707}{16}$$

Acorde con el ejercicio planteado se muestran los siguientes cálculos

1700000*41,02% = 69734 –> Resto * Tipo marginal

1700000*38,86% = 66062 –> Resto * Tipo marginal (El que desea el usuario)

69734 – 66062 = 3672 –> Diferencia

Con el primer caso (interpolación de segundo grado), reemplazando el valor de base imponible (5000000) en el polinomio se obtuvo:

```
> 155.125 + 0.1509214*(5000000) + 2.572279e-08*(5000000)^2
[1] 1397832
```

Comparando con el valor original de pago (1398924), solo hay 1092 de desigualdad.

Para el segundo caso (interpolación de tercer grado), realizando lo anterior:

```
> 38707/16 + 167449*(5000000)/1120000 + 6119*(5000000)**2/235200000000 - (5000000)**3/49392000000000000
[1] 1397833
```

La diferencia con el valor de pago original es de 1091.

Conclusiones

Después de lo expuesto, se llegó a la siguiente respuesta: el método óptimo para calcular la imposición correspondiente a la base de 5 millones es interpolación de 2 grado, debido a que maneja un ahorro no mucho mayor. No fue posible cumplir con las demandas del usuario, sin embargo, se logró casi un 30% de lo pedido.