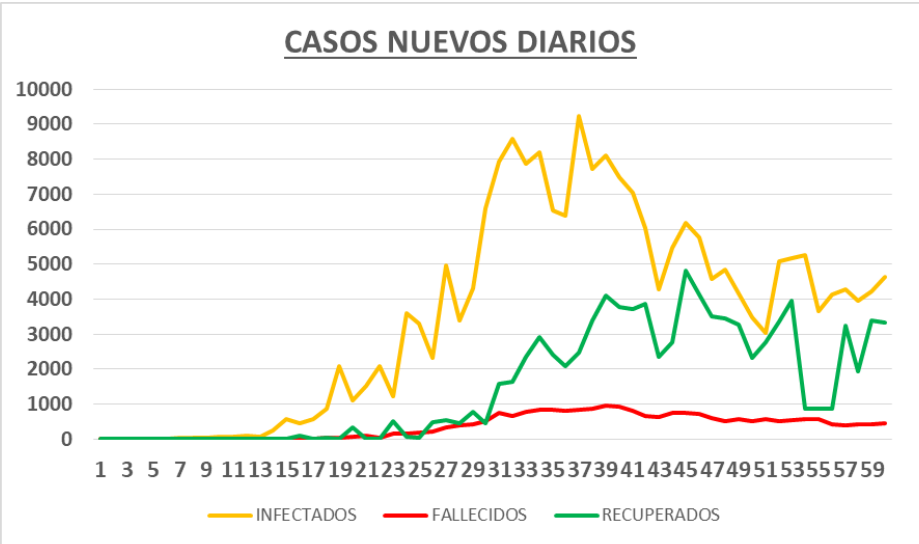
Según el Gobierno de España, la epidemia causada por la COVID 19 en España, durante los 45 días que comienzan el 24 de febrero de 2020, ha provocado los **Infectados nuevos** diarios (**IN**), los **fallecidos nuevos** diarios (**FN**) y los **recuperados nuevos** diarios (**RN**) que se recogen en la tabla siguiente:

	CASOS DIARIOS		
DÍA	**INFECTADOS**	**FALLECIDOS**	**RECUPERADOS**
1	1	0	0
2	4	0	0
3	6	0	0
4	12	0	0
5	19	0	0
6	14	0	0
7	25	0	0
8	41	0	0
9	44	1	0
10	59	1	1
11	54	0	2
12	83	5	7
13	65	2	7
14	244	7	0
15	557	13	13
16	464	6	105
17	583	19	3
18	869	31	51
19	2086	47	4
20	1100	60	324
21	1512	99	0
22	2098	50	13
23	1236	149	498
24	3591	147	53
25	3308	195	26
26	2333	210	481
27	4964	335	537
28	3394	394	450
29	4321	410	780
30	6584	514	439
31	7937	738	1573
32	8578	655	1648
33	7871	769	2342
34	8189	836	2928
35	6549	838	2424
36	6398	808	2071

	CASOS DIARIOS		
DÍA	INFECTADOS	FALLECIDOS	RECUPERADOS
37	9222	849	2479
38	7719	864	3388
39	8102	950	4096
40	7472	932	3770
41	7026	809	3706
42	6023	674	3861
43	4273	637	2357
44	5478	743	2771
45	6180	757	4813
46	5756	728	4144
47	4576	605	3503
48	4830	510	3441
49	4167	574	3282
50	3477	517	2336
51	3045	567	2777
52	5092	523	3349
53	5183	551	3944
54	5252	585	853
55	3658	565	853
56	4128	410	854
57	4266	399	3230
58	3968	430	1927
59	4211	435	3401
60	4635	440	3335

Si se representan los datos de nuevos casos se tienen curvas como las siguientes que muestran una gran irregularidad:



Las autoridades sanitarias también proporcionan las cifras acumuladas. En ese sentido la mera suma, día tras día, de los casos de nuevos infectados (que son los datos proporcionados) no coincide con el número de infectados existentes[1] pues no excluye de ese grupo ni a los recuperados ni a los fallecidos. Por ello lo primero que se te pide es:

A) Escribe un subprograma function en **MATLAB**, llamado **EXISTENTES**, en el que sean **argumentos de entrada** los vectores **IN**, **RN** y **FN** de la tabla anterior, y llamando **ND** al número de elementos de cada uno de dichos vectores (es decir, con la tabla anterior **ND** será **60**) calcule como **argumentos de salida** los vectores **IE**, **RE** y **FE** obtenidos mediante las expresiones:

$$IE_1 = 4;\ RE_1 = 0;\ FE_1 = 1;$$

$$IE_J = IE_{J-1} + IN_J - RN_J - FN_J; \qquad (J=2,\ldots,ND)$$

$$RE_J = RE_{J-1} + RN_J; \qquad (J=2,\ldots,ND)$$

$$FE_J = FE_{J-1} + FN_J; \qquad (J=2,\ldots,ND)$$

B) En lo que sigue nos será de interés tener una función que resuelva sistemas de ecuaciones tridiagonales, es decir sistemas de la forma:

$$\begin{pmatrix} B_1 & C_1 & 0 & \cdots & \cdots & 0 & \cdots & \cdots & 0 \\ A_2 & B_2 & C_2 & \cdots & \cdots & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \cdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & A_J & B_J & C_J & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & \cdots & 0 & A_{N-1} & B_{N-1} & C_{N-1} \\ 0 & \cdots & 0 & \cdots & \cdots & \cdots & 0 & A_N & B_N \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_J \\ \vdots \\ \vdots \\ x_{N-1} \\ x_N \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ \vdots \\ F_J \\ \vdots \\ \vdots \\ F_{N-1} \\ F_N \end{pmatrix}$$

Una manera eficaz de resolver este tipo de sistemas es el denominado algoritmo de Thomas[2] que se resume en:

---

[1] *Eso sin entrar en la mala calidad de los datos proporcionados pues no contabilizan ni aproximadamente las cifras más probables de infectados no detectados por falta de pruebas para ello, o de recuperados que fueron asintomáticos o simplemente no diagnosticados, y, ni tan siquiera, la cifra de fallecidos reales. Por citar un ejemplo de la pobre calidad de los datos, la cifra de recuperados acumulados que proporciona el ministerio presenta los días 17 y 18 de abril valores inferiores a los del 16 de abril, lo cual no tiene sentido. Por eso en la tabla se han modificado los datos del 17, 18 y 19 de abril.*

[2] *El algoritmo que se describe fue diseñado el físico-matemático británico* por Llewellyn Hilleth **Thomas** *(Londres, 1903 – Raleigh (Carolina del Norte), 1992) como una adaptación a los sistemas tridiagonales del método de eliminación gaussiana. De esa forma se realizan del orden de (8·N) operaciones mientras que la aplicación del método de Gauss directamente exigiría realizar del orden de ($2 \cdot N^3/3$) operaciones. Así en el caso N=46 el algoritmo de Thomas realiza del orden de 368 operaciones para determinar la solución mientras que el uso del método de Gauss conllevaría unas 6500 operaciones.*

a) En primer lugar se calcula un vector llamado **r** con **(N-1)** elementos y en el que:

$$r_1 = C_1/B_1, \qquad r_J = C_J/(B_J - r_{J-1} \cdot A_J) \qquad (2 \le J \le (N-1))$$

b) Tras ello se calculará un nuevo vector **z** de **N** elementos dados por:

$$z_1 = F_1/B_1, \qquad z_J = (F_J - z_{J-1} \cdot A_J)/(B_J - r_{J-1} \cdot A_J) \quad (2 \le J \le N)$$

c) Por último, se obtendrá el vector solución **x** mediante:

$$x_n = z_n, \qquad x_J = z_J - r_J \cdot x_{J+1} \quad (J = N-1, \ldots, 1)$$

Por ello SE PIDE que escribas un subprograma **function**, llamado **ATH**, que tenga como argumentos de entrada cuatro vectores llamados **A**, **B,C** y **F** con los que se calculará como argumento de salida un vector **x** aplicando el algoritmo de Thomas que se acaba de describir.

Con el objeto de suavizar más las curvas de Infectados Existentes (**IE**), Recuperados Existentes (**RE**) y Fallecidos Existentes (**FE**), procederemos a calcular las funciones **spline cúbica** que las interpolan.

La función spline cúbica interpoladora, que denominaremos **u(x)**, de una tabla de puntos (J, F_J) **(J=1,…,N)** es una función continua en el intervalo **(1,ND)**, con sus dos primeras derivadas también continuas en dicho intervalo, que pasa por todos los puntos y que está definida por **(ND-1)** tramos polinómicos de grado menor o igual que 3:

$$u(x) = \begin{cases} u^{(1)}(x) = a_1 + b_1 \cdot (x-1) + c_1 \cdot (x-1)^2 + d_1 \cdot (x-1)^3 & 1 \le x < 2 \\ u^{(2)}(x) = a_2 + b_2 \cdot (x-2) + c_2 \cdot (x-2)^2 + d_2 \cdot (x-2)^3 & 2 \le x < 3 \\ \ldots\ldots\ldots\ldots\ldots\ldots & \ldots\ldots \\ u^{(J)}(x) = a_J + b_J \cdot (x-J) + c_J \cdot (x-J)^2 + d_J \cdot (x-J)^3 & J \le x < J+1 \\ \ldots\ldots\ldots\ldots\ldots\ldots & \ldots\ldots \\ u^{(ND-1)}(x) = a_{ND-1} + b_{ND-1}(x-(ND-1)) + c_{ND-1}(x-(ND-1))^2 + d_{ND-1}(x-(ND-1))^3 & ND-1 \le x \le ND \end{cases}$$

El cálculo de los **4·(ND-1) coeficientes** a_J, b_J, c_J y d_J (**J=1, …,ND-1**) con los que se definen los (ND-1) tramos es el objeto del tercer subprograma **function** que se te pide en este ejercicio (y puedes consultar el proceso mediante el que se obtienen las expresiones que se utilizan, en el **ANEXO 1** de este ejercicio).

C) Escribe un subprograma **function** en **MATLAB**, llamado **SPL3**, que teniendo como argumento de entrada un vector llamado **F**, de **N** elementos, calcule como argumentos de salida **4** vectores llamados **a**, **b**, **c** y **d**, cada uno de ellos con **(N-1)** elementos, calculados de la forma siguiente:
Los elementos del vector **a** se obtendrán mediante:

$$a_J \leftarrow F_J \qquad (J = 1, 2, 3, \ldots, (N-1))$$

Los elementos del vector **c** se obtendrán resolviendo el sistema de ecuaciones:

$$\begin{pmatrix} 2 & 1 & 0 & \cdots & \cdots & 0 & 0 \\ 1 & 4 & 1 & \ddots & \cdots & & 0 \\ 0 & 1 & 4 & \ddots & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 4 \end{pmatrix} \cdot \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ \vdots \\ c_{N-2} \\ c_{N-1} \end{Bmatrix} = 3 \cdot \begin{Bmatrix} F_2 - F_1 \\ F_1 - 2 \cdot F_2 + F_3 \\ F_2 - 2 \cdot F_3 + F_4 \\ \vdots \\ \vdots \\ F_{N-3} - 2 \cdot F_{N-2} + F_{N-1} \\ F_{N-2} - 2 \cdot F_{N-1} + F_N \end{Bmatrix}$$

Resuelve el sistema anterior utilizando el ***algoritmo de Thomas*** que se programó en el apartado anterior.

Tras ello, los elementos del vector **d** se calcularán mediante las expresiones:

$$d_{J-1} \leftarrow \frac{c_J - c_{J-1}}{3} \qquad (J = 2, 3, \ldots, N-1); \qquad d_{N-1} \leftarrow \frac{-c_{N-1}}{3}$$

Finalmente los elementos del vector **b** tendrán el valor:

$$b_{J-1} \leftarrow F_J - F_{J-1} - c_{J-1} - d_{J-1} \qquad (J = 2, 3, \ldots, N)$$

D) Escribe un programa **MATLAB** que:

D-1) partiendo de la tabla dada construya dos subventanas de dibujo horizontales, representando en la primera de ellas las curvas de evolución diaria de Infectados nuevos (**IN**), Recuperados Nuevos (**RN**) y Fallecidos Nuevos (**FN**), y

D-2) usando las funciones anteriores calcule las cifras de Infectados Existentes (**IE**), Recuperados Existentes (**RE**) y fallecidos Existentes (**FE**), así como las curvas spline cúbicas que interpolan a las nubes de puntos:

D-2-1) $(J,\ IE_J) \quad (J=1, \ldots, ND)$

D-2-2) $(J,\ RE_J) \quad (J=1, \ldots, ND)$

D-2-3) $(J,\ FE_J) \quad (J=1, \ldots, ND)$

D-3) tras ello representa en la segunda de las subventanas de dibujo, usando colores y estilos de línea diferentes que las identifiquen, las nubes de puntos (J, IE_J), (J, RE_J) y (J, FE_J) $(J=1, \ldots, ND)$, así como las funciones spline cúnicas con las que se las interpola.

ANEXO 1: Obtención de los coeficientes de la spline cúbica pedida en el apartado B)

Para determinar la función spline cúbica $u(x)$ *que en las abscisas* $\{1,2,3,4,\dots,N\}$ *toma los valores* $\{F_1,\ F_1,\ F_3,\ F_4,\dots,F_N\}$*, expresaremos dicha función mediante:*

$$u(x) = \begin{cases} u^{(1)}(x) = a_1+b_1\cdot(x-1)+c_1\cdot(x-1)^2+d_1\cdot(x-1)^3 & 1\le x<2 \\ u^{(2)}(x) = a_2+b_2\cdot(x-2)+c_2\cdot(x-2)^2+d_2\cdot(x-2)^3 & 2\le x<3 \\ \dots\dots\dots\dots\dots\dots & \dots\dots \\ u^{(J)}(x) = a_J+b_J\cdot(x-J)+c_J\cdot(x-J)^2+d_J\cdot(x-J)^3 & J\le x<J+1 \\ \dots\dots\dots\dots\dots\dots & \dots\dots \\ u^{(ND-1)}(x)=a_{ND-1}+b_{ND-1}\left(x-(ND-1)\right)+c_{ND-1}\left(x-(ND-1)\right)^2+d_{ND-1}\left(x-(ND-1)\right)^3 & ND-1\le x\le ND \end{cases}$$

necesitamos conocer los $4\cdot(N-1)$ ***coeficientes*** a_J, b_J, c_J *y* d_J ($J=1,\ \dots,N-1$) *de cada tramo. Para ello se deben plantear* $4\cdot(N-1)$ ***ecuaciones*** *que nos las proporcionen. Entre ellas se incluyen las siguientes:*

**) Los* N *valores que debe tomar la función cada día:*

$$u(J) = F_J \ (J=1,2,\dots,N) \quad \text{(Ecs. 1)}$$

**) Las* $(N-2)$ *ecuaciones resultantes de igualar en las abscisas* $x=J$, con $J=2,\dots,(N-1)$*, los valores del tramo* $(J-1)$ *y del tramo* J*:*

$$u^{(J-1)}(J)=u^{(J)}(J) \quad (J=2,\dots,N-1) \qquad \text{(Ecs. 2)}$$

) Las* $(N-2)$ *ecuaciones resultantes de igualar en las abscisas* $x=J$, con $J=2,\dots,(N-1)$, *los* ***valores de la primera derivada* *del tramo* $(J-1)$ *y del tramo* J*:*

$$\left(u^{(J-1)}\right)'(J) = \left(u^{(J)}\right)'(J) \qquad \left(J=2,3,\dots,(N-1)\right) \qquad \text{(Ecs. 3)}$$

) Las* $(N-2)$ *ecuaciones resultantes de igualar en las abscisas* $x=J$*, con* $J=2,\dots,(N-1)$, *los* ***valores de la segunda derivada* *del tramo* $(J-1)$ *y del tramo* J*:*

$$\left(u^{(J-1)}\right)''(J) = \left(u^{(J)}\right)''(J) \qquad \left(J=2,3,\dots,(N-1)\right) \qquad \text{(Ecs. 4)}$$

Con ello se tendrán $N+3(N-2)=4N-6$*, faltándonos aún otras dos condiciones adicionales. En el caso de la función spline que se calcula en este ejercicio, se ha optado por imponer las* ***dos condiciones adicionales*** *siguientes:*

$$u'(1)=0 \quad \text{(Ec.5)} \qquad y \qquad u''(N)=0 \quad \text{(Ec.6)}.$$

Manipulemos un poco las ecuaciones anteriores.

PRIMERO: *Del grupo de ecuaciones* **(Ecs. 1)** *las* **(N-1)** *primeras nos proporcionan las igualdades:*

$$u(J) = IE_J \Leftrightarrow a_J = F_J \qquad (J=1,2,3,...,N-1) \qquad \text{(COEFS. a)}$$

en tanto que la última de la ecuaciones del grupo **(Ecs.1)** *se puede escribir como:*

$$u(N) = F_N \Leftrightarrow b_{N-1}+c_{N-1}+d_{N-1} = F_N-F_{N-1} \Rightarrow b_{N-1}=F_N-F_{N-1}-c_{N-1}-d_{N-1}$$

*Esta última ecuación nos la vamos a "*reservar*" para añadirla a las del grupo siguiente:*

SEGUNDO: *Del grupo de ecuaciones* **(Ecs.2)** *obtenidas imponiendo la continuidad de la función* **u(x)***, se infieren las igualdades:*

$$u(J-1)(J)=u(J)(J) \qquad (J=1,2,\ldots,N) \Rightarrow$$
$$\Rightarrow b_{J-1} + c_{J-1} + d_{J-1} = F_J-F_{J-1} \qquad (J=2,3,...,N-1) \Rightarrow$$
$$\Rightarrow b_{J-1} = F_J-F_{J-1}-c_{J-1}-d_{J-1} \qquad (J=2,3,...,N-1)$$

que, unida con la que reservamos del grupo anterior, nos permite escribir:

$$b_{J-1} = F_J-F_{J-1}-c_{J-1}-d_{J-1} \qquad (J=2,3,...,N) \qquad \text{(COEFS. b)}$$

Las ecuaciones anteriores nos proporcionarán los coeficientes $\mathbf{b_1}$,$\mathbf{b_2}$, …,$\mathbf{b_{N-2}}$ *y* $\mathbf{b_{N-1}}$ *pero deben ser utilizadas sólo cuando se conozcan los coeficientes* $\mathbf{c_1}$,$\mathbf{c_2}$, …,$\mathbf{c_{N-2}}$ *y* $\mathbf{d_1}$,$\mathbf{d_2}$, …,$\mathbf{d_{N-2}}$, *cosa de la que nos ocuparemos a continuación.*

TERCERO: *Del grupo de ecuaciones* **(Ecs.4)** *obtenidas imponiendo la continuidad de la* ***segunda*** *derivada inferimos que:*

$$c_{J-1}+3 \cdot d_{J-1} = c_J \Rightarrow d_{J-1}= \frac{c_J-c_{J-1}}{3} \qquad (J=2,3,...,N-1)$$

Estas ecuaciones nos proporcionan los coeficientes $\mathbf{d_1}$,$\mathbf{d_2}$, …,$\mathbf{d_{N-2}}$ *por lo que podrían ya ser utilizados en el cálculo de los coeficientes* (**COEFS b**). *Pero nos falta aún el coeficiente* $\mathbf{d_{N-1}}$ *por determinar.*

Para obtener $\mathbf{d_{N-1}}$ *utilizaremos la* **(Ec.6)** *de la que se infiere que:*

$$u''(N) = 0 \Leftrightarrow c_{N-1}+3 \cdot d_{N-1} = 0 \Leftrightarrow d_{N-1} = \frac{-c_{N-1}}{3}$$

En resumen:

$$d_{J-1}= \frac{c_J-c_{J-1}}{3} \qquad (J=2,3,...,N-1); \qquad d_{N-1}= \frac{-c_{N-1}}{3} \qquad \text{(COEFS. d)}$$

Las ecuaciones anteriores nos proporcionarán los coeficientes $\mathbf{d_1}$,$\mathbf{d_2}$, …,$\mathbf{d_{N-2}}$ *y* $\mathbf{d_{N-1}}$ *pero deben ser utilizadas sólo cuando se conozcan los coeficientes* $\mathbf{c_1}$,$\mathbf{c_2}$, …,$\mathbf{c_{N-1}}$ *de lo cual nos ocupamos a continuación*

__CUARTO:__ Del grupo de ecuaciones **(Ecs.4)** *obtenidas imponiendo la continuidad de la* ***primera*** *derivada inferimos que:*

$$b_{J-1} + 2\cdot c_{J-1} + 3\cdot d_{J-1} = b_J \qquad (J=2,3,...,N-1) \Rightarrow$$
$$\Rightarrow 2\cdot c_{J-1} + 3\cdot d_{J-1} = b_J - b_{J-1} \qquad (J=2,3,...,N-1)$$

Que, utilizando las expresiones **(COEFS b)** *podemos rescribir como:*

$$2\cdot c_{J-1}+3\cdot d_{J-1} = (F_{J+1}-F_J-c_J-d_J) - (F_J-F_{J-1}-c_{J-1}-d_{J-1}) \qquad (J=2,3,...,N-1) \Rightarrow$$
$$\Rightarrow 2\cdot c_{J-1}+3\cdot d_{J-1} = F_{J-1}-2\cdot F_J+F_{J+1}-c_J+c_{J-1}-d_J+d_{J-1} \qquad (J=2,3,...,N-1) \Rightarrow$$
$$\Rightarrow c_{J-1}+c_J + 2\cdot d_{J-1} + d_J = (F_{J-1}-2\cdot F_J+F_{J+1}) \qquad (J=2,3,...,N-1)$$

Utilizando ahora la expresiones (**COEFS d**) *se plantean par los valores* **J=2,3,…,N-2** *las ecuaciones:*

$$c_{J-1}+c_J+2\cdot\frac{c_J-c_{J-1}}{3}+\frac{c_{J+1}-c_J}{3} = (F_{J-1}-2\cdot F_J+F_{J+1}) \qquad (J=2,3,...,N-2) \Rightarrow$$
$$\Rightarrow c_{J-1}+4c_J+c_{J+1}= 3\cdot(F_{J-1}-2\cdot F_J+F_{J+1}) \qquad (J=2,3,...,N-2)$$

Asimismo, la expresión de d_{N-1} *escrita en* (**COEFS. d**) *nos permite obtener:*

$$c_{N-2}+c_{N-1}+ 2\cdot d_{N-2} + d_{N-1} = (F_{N-2}-2\cdot F_{N-1}+F_N) \Rightarrow$$
$$\Rightarrow c_{N-2}+c_{N-1}+ 2\cdot\frac{c_{N-1}-c_{N-2}}{3} - \frac{c_{N-1}}{3} = (F_{N-2}-2\cdot F_{N-1}+F_N) \Rightarrow$$
$$\Rightarrow c_{N-2}+4\cdot c_{N-1}= 3\,(F_{N-2}-2\cdot F_{N-1}+F_N)$$

Se tienen así **(N-2)** *ecuaciones, que involucran a los* **(N-1)** *coeficientes* $c_1, c_2, \ldots, c_{N-1}$ *faltándonos aun por imponer la que se deduce de la* **(Ec. 5)**:

$$u'(1) = 0 \Leftrightarrow b_1 = 0$$

que, habida cuenta de la expresión de b_1 *que proporciona* **(COEFS b)** *rescribiremos como:*

$$b_1 = 0 \Leftrightarrow F_2-F_1-c_1-d_1 = 0 \Leftrightarrow c_1 + d_1 = F_2 - F_1$$

Usando la expresión de d_1 *proporcionada por* **(COEFS d)** *finalmente se obtiene:*

$$c_1 + d_1 = F_2-F_1 \Rightarrow c_1 + \frac{c_2-c_1}{3} = F_2-F_1 \Rightarrow 2c_1+c_2 = 3\cdot(F_2-F_1)$$

En resumen, los coeficientes $c_1, c_2, \ldots, c_{N-1}$ *se obtienen resolviendo el sistema de ecuaciones:*

$$\begin{pmatrix} 2 & 1 & 0 & \cdots & \cdots & 0 & 0 \\ 1 & 4 & 1 & \ddots & \cdots & & 0 \\ 0 & 1 & 4 & \ddots & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 1 & 4 \end{pmatrix} \cdot \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ \vdots \\ c_{N-2} \\ c_{N-1} \end{Bmatrix} = 3\cdot \begin{Bmatrix} F_2-F_1 \\ F_1-2\cdot F_2+F_3 \\ F_2-2\cdot F_3+F_4 \\ \vdots \\ \vdots \\ F_{N-3}-2\cdot F_{N-2}+F_{N-1} \\ F_{N-2}-2\cdot F_{N-1}+F_N \end{Bmatrix} \qquad \textbf{(COEFS. c)}$$