JAVIER MARCOS
Actuario Director Departamento Fondos de Pensiones. La Unión y el Fénix

# Ampliación de las aplicaciones actuariales en calculadoras financieras

*En el número anterior de nuestra revista, el autor nos abría las puertas para la utilización de las calculadoras financieras en fórmulas actuariales. Reconocida su utilidad y ahorro de tiempo, que contrastan con su facilidad de programación, nos ha hecho solicitarle que amplíe a nuevos supuestos su trabajo anterior.*

EN primer lugar, agradezco todas las llamadas que he recibido en relación con la publicación anterior. Sólo se trataba de aportar alguna idea, y demostrar que poseemos suficientes herramientas de manejo muy sencillo para desarrollar nuestra labor como actuarios.

Lo que se planteaba casi como un juego, ha determinado un gran interés por la calculadora utilizada, el modelo 19B II de Hewlett Packard, y por el contenido de la fórmula programada. Por eso, me veo obligado a explicar de forma somera no solamente ésta, sino los pasos seguidos en su programación.

En mi opinión, se expone un instrumento más bien de apoyo, de comprobación de los cálculos realizados en un ordenador.

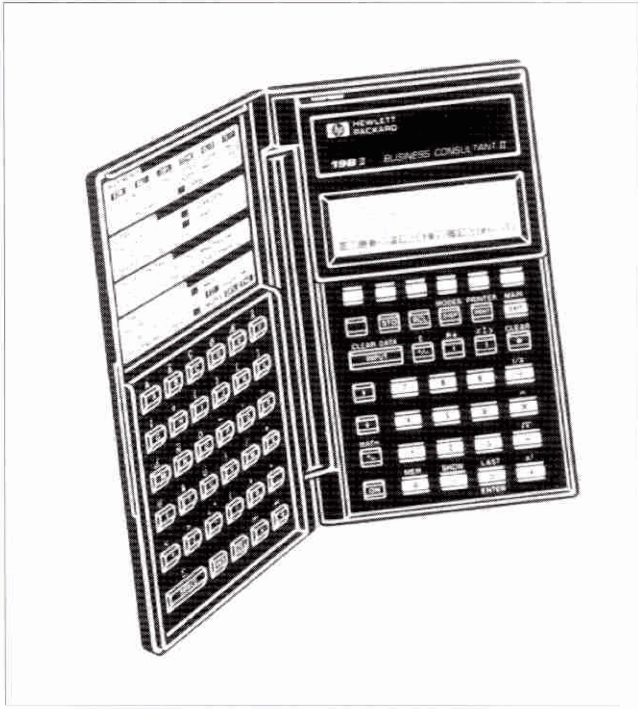
En cualquier caso, vamos a ir un poco más allá, ya que introduciremos alguna mejora para dar más posibilidades al programa, y aprenderemos a conectarlo con otros desarrollos.

En las próximas líneas vamos a exponer tres aspectos:

— Planteamiento de la fórmula.

— Visión crítica de su programación. Mejoras.

— Nuevos desarrollos y conexiones entre los programas.




## I. Planteamiento de la fórmula

Como siempre, debemos distinguir lo que queremos realizar de cómo lo realizamos. Quienes utilicen otro tipo de herramientas distintas de las aquí propuestas, podrán muy bien quedarse con este planteamiento y desarrollarlo en otro tipo de entornos.

La fórmula propuesta se puede encontrar, de manera más o menos explícita, en algunos manuales de matemática actuarial. Sin embargo, ha sido la práctica y el contacto con otros profesionales por lo que puedo ofrecer esta pequeña aportación.

Una renta actuarial constante, vitalicia y pospagable se puede plantear de la siguiente manera:

$$\text{Renta Pospagable} = \sum_{t=1}^{w-x} \left[ v^{t-1} \frac{l_{x+t-1}}{l_x} \frac{m-1}{2m} + v^{t} \frac{l_{x+t}}{l_x} \frac{m+1}{2m} \right]$$

Donde m indica el fraccionamiento, 12 en el caso de mensual, 4 trimestral, 3 cuatrimestral y 1 anual.

Si en lugar de ser vitalicia fuera temporal n años, el sumatorio llegaría hasta el máximo entre n y w-x.

Por otra parte, si en lugar de ser pospagable fuera prepagable, intercambiaríamos los términos m-1 y m+1. Es decir,

$$\text{Renta Prepagable} = \sum_{t=1}^{s} \left[ v^{t-1} \frac{l_{x+t-1}}{l_x} \frac{m+1}{2m} + v^{t} \frac{l_{x+t}}{l_x} \frac{m-1}{2m} \right]$$

Donde s= MAX(n, w-x)

Si necesitamos obtener el valor de una renta creciente en progresión geométrica, pospagable, haríamos:

$$R = \sum_{t=1}^{s} (1+c)^{t-1} \left[ v^{t-1} \frac{l_{x+t-1}}{l_x} \frac{m-1}{2m} + v^{t} \frac{l_{x+t}}{l_x} \frac{m+1}{2m} \right]$$

Donde c representa el crecimiento anual en tanto por uno.

Si el crecimiento no es geométrico sino aritmético, la fórmula, en el caso de una renta prepagable, sería:

$$R = \sum_{t=1}^{s} (1+(t-1)c) \left[ v^{t-1} \frac{l_{x+t-1}}{l_x} \frac{m+1}{2m} + v^{t} \frac{l_{x+t}}{l_x} \frac{m-1}{2m} \right]$$

Este planteamiento se puede ampliar al caso de rentas de dos cabezas, añadiendo únicamente la probabilidad de supervivencia de la segunda. Vamos a usar la letra «y» para mencionar la edad actuarial de esta segunda cabeza.

Si hacemos:

$$R = \sum_{t=1}^{s} (1+c)^{t-1} \left[ v^{t-1} \frac{l_{x+t-1}}{l_x} \frac{l_{y+t-1}}{l_y} \frac{m-1}{2m} + v^{t} \frac{l_{x+t}}{l_x} \frac{l_{y+t}}{l_y} \frac{m+1}{2m} \right]$$

Obtenemos, dando valores a las distintas variables, cualquier combinación de renta de dos cabezas pospagable y creciente en progresión geométrica. Si lo que nos interesa es que sea prepagable y creciente en progresión aritmética:

$$R = \sum_{t=1}^{s} (1+(t-1)c) \left[ v^{t-1} \frac{l_{x+t-1}}{l_x} \frac{l_{y+t-1}}{l_y} \frac{m+1}{2m} + v^{t} \frac{l_{x+t}}{l_x} \frac{l_{y+t}}{l_y} \frac{m-1}{2m} \right]$$

Por último, este planteamiento tan útil para hallar el valor de una renta discreta (con las aproximaciones que se realizan en la práctica), se puede ampliar a las rentas financieras. Unicamente debemos sustituir todo lo referente a lx, ly por 1.

Así, una renta financiera creciente en progresión geométrica, prepagable, temporal n años se puede plantear como:

$$R = \sum_{t=1}^{n} (1+(t-1)c) \left[ v^{t-1} \frac{m+1}{2m} + v^{t} \frac{m-1}{2m} \right]$$

Dejo al lector que piense en las posibilidades de este planteamiento cuando V=1. Estos valores se pueden hallar con el programa. Simplemente I%=0.

## II. Visión crítica de su programación. Mejoras

Hasta el momento lo único que hemos hecho ha sido sistematizar un planteamiento que no es novedoso.

A partir de este punto, vamos a explicar el programa desarrollado en el número anterior de la revista y vamos a introducir una pequeña mejora para ampliar sus posibilidades.

La forma elegida en el programa ha sido la siguiente:

$$R = \frac{1}{l_x\, l_y\, 2m} \sum_{t=1}^{s} FC \left[ v^{t-1}\, l_{x+t-1}\, FASE + v^{t}\, l_{x+t}\, FASE \right]$$

Como se puede comprobar, se ha puesto fuera del sumatorio todas aquellas variables que no son afectadas por el mismo. La razón es muy sencilla. Si programásemos la fórmula de la manera expuesta en el apartado anterior, la calculadora tendría que realizar más operaciones. Por tanto, tardaría más tiempo en llegar a la solución.

El segundo punto importante, aparte de la forma que adopta la fórmula, es pensar en las situaciones con las que nos podemos encontrar. Se han previsto, respecto a x e y, cuatro posibilidades, que son la columna vertebral del programa:

a) Que x e y sean mayor que cero
b) Solamente x es mayor que cero
c) Solamente y es mayor que cero
d) Ambas variables son igual a cero

Comencemos a ver cada línea:

*Primer bloque:* El término anterior al sumatorio se desarrolla en las cinco primeras líneas. Nótese que éstas se presentan tal y como se exponen en la pantalla de la calculadora.

```
RENTA=IF(X>0 AND Y>0:I
NV(ITEM(M80:X)xITEM(F80
:Y)):IF(X>0:INV(ITEM(M8
0:X)):IF(Y>0:INV(ITEM(F
80:Y)):1)))xINV(2xM)
```

a) En el caso de que x e y sean variables positivas, debemos hacer $(1/l_x)$ x $(1/l_y)$. Por tanto:

IF(X>0 AND Y>0:INV(ITEM(M80:X)xITEM(F80:Y)):

b) Si solamente x mayor que cero:
IF(X>0:INV(ITEM(M80:X)):

c) Si solamente y mayor que cero:
IF(Y>0:INV(ITEM(F80:Y)):

d) Si ambas son cero hacemos que el valor sea uno: 1)))

Por último, solamente nos resta multiplicar por el término 1/(2xM). Por eso hacemos: xINV(2xM)x

No necesariamente este punto de vista tiene que ser el más satisfactorio. Algunos preferirán, en el caso d), que no se calcule nada y advertir de esta forma un posible error en la introducción de los datos. Para ello, se puede poner un 0 en lugar de 1.

Por último, advertir que nos da igual utilizar la función INV que programar una simple división. Va en gustos. Tendría importancia si se comprobara que la velocidad de cálculo varía de forma apreciable usando una u otra forma.

El mismo comentario cuando tengamos que dividir una variable entre 100. Habrá quien prefiera multiplicar por 0,01. Vamos a hacerlo así, con objeto de aprender un detalle más.

*Segundo bloque:*

Llegamos al sumatorio. Le indicamos que la variable que va a actuar de contador es T, y que su valor comienza en uno. Es decir estamos haciendo: $\sum_{t=1}$

Para ello, introducimos en la calculadora: Σ(T:1:

¿Hasta dónde debe llegar el sumatorio? De nuevo nos encontramos con las situaciones a, b, c, d expuestas:

a) Debe llegar hasta el mínimo de tres posibilidades: w-x, w-y o n. Por eso:

IF(X>0 AND Y>0:MIN(MIN(SIZES(M80)-X:SIZES(F80)-Y):N):

b) Hasta el mínimo entre w-x y n:
IF(X>0:MIN(SIZES(M80)-X:N):

c) Hasta el mínimo entre w-y y n:
IF(Y>0:MIN(SIZES(F80)-Y:N):

c) Hasta n. Aquí se encuentra la mejora. Sustituimos en las líneas 10 y 11 la expresión SIZES(F80)-MAX(X:Y) por N. Así de sencillo. De esta manera, podemos calcular todo tipo de rentas financieras temporales.

Por último, decimos a la calculadora que T, la variable sumatorio, va incrementándose de uno en uno: :1:

Este segundo bloque, con la mejora, quedaría así:

```
xΣ(
T:1:IF(X>0 AND Y>0:MIN(
MIN(SIZES(M80)-x:SIZES(
F80)-Y):N):IF(X>0:MIN(S
IZES(M80)-X:N):IF(Y>0:M
IN(SIZES(F80)-Y:N):N)))
:1:
```

*Tercer bloque*, y último. Entramos en el sumatorio de la fórmula planteada al inicio.

Dividimos este tercer bloque en tres partes:

1. Planteamiento del crecimiento de la renta. Puede ser constante, creciente en progresión geométrica o aritmética. La forma constante se puede obtener en función de cualquiera de estas dos últimas, con un valor de c igual a cero. Por tanto, lo único que debemos decir a la calculadora es si queremos que crezca la renta de forma geométrica o aritmética. Para ello, elegimos otra variable, FC (forma de crecimiento). Si toma valor igual a uno realizaría: (1+c)t-1. En caso contrario: (1+(t-1)c)

Como podemos ver, este no deja de ser un criterio, una forma de dar solución al problema planteado. Por eso, se ha programado:

IF(FC=1:(1+C%x.01)^(T-1):(1+C%x.01x(T-1)))

Donde C% es el nombre de la variable c expuesta al comienzo. Este nombre nos ayuda a recordar que el valor debe ser un número tal como 8, 9 ... y que el programa se encarga de dividir entre 100 (multiplicando por 0,001) y sumarle 1.

Esta parte se recoge en nuestra pantalla así:

```
IF(FC=1:(1+C%x.01)^(
T-1):(1+C%x.01x(T-1)))x
(INV(1+I%x.01)^(T-1)
```

2. Primer término de la suma:

Vamos a hallar $v^{t-1}$. Escribimos en la calculadora:

x(INV(1+I%x.01)^(T-1)x

Ahora introducimos las probalidades. De nuevo nos enfrentamos con las posibilidades a, b, c, d.

a) Hacemos $l_{x+t-1}$ $l_{y+t-1}$. Por eso:

IF(X>0 AND Y>0:ITEM(M80:X+T-1)xITEM(F80:Y+T-1):

b) Unicamente $l_{x+t-1}$: IF(X>0:ITEM(M80:X+T=1):

c) Unicamente $l_{y+t-1}$: IF(Y>0:ITEM(F80:Y+T-1):

d) De nuevo, ponemos uno: 1)))

Cabe el mismo comentario realizado anteriormente. Podríamos poner un cero, aunque en este caso sería redundante con el que se hubiera puesto en el bloque primero.

Unicamente nos queda por introducir m+1 en caso de prepagable y m-1 en caso de ser pospagable. Usemos el mismo criterio empleado con la variable FC. Creamos una variable que denominamos FASE. Si ésta toma valor igual a uno, el primer término de la suma se multiplicará por m+1. En caso contrario, por m-1. Hacemos: xIF(FASE=1:M+1:M-1)

En pantalla queda:

```
                     xIF
(X>0 AND Y>0:ITEM(M80:X
+T-1)xITEM(F80:Y+T-1):I
F(X>0:ITEM(M80:X+T-1):I
F(Y>0:ITEM(F80:Y+T-1):1
)))xIF(FASE=1:M+1:M-1)
```

2. Segundo término de la suma. El mismo esquema:

Hallamos $v^t$: + (INV(1+I%x.01)^(T)x

Los paréntesis de la T podrían sobrar. En este caso los ponemos para seguir exactamente el mismo esquema anterior.

Introducimos las probabilidades:

a) $l_{x+t}\ l_{y+t}$

IF(X>0 AND Y>0:ITEM(M80:X+T-1)xITEM(F80:Y+T):

b) $l_{x+t}$: IF(X>0:ITEM(M80:X+T):

c) $l_{y+t}$: IF(Y>0:ITEM(F 80:Y+T):

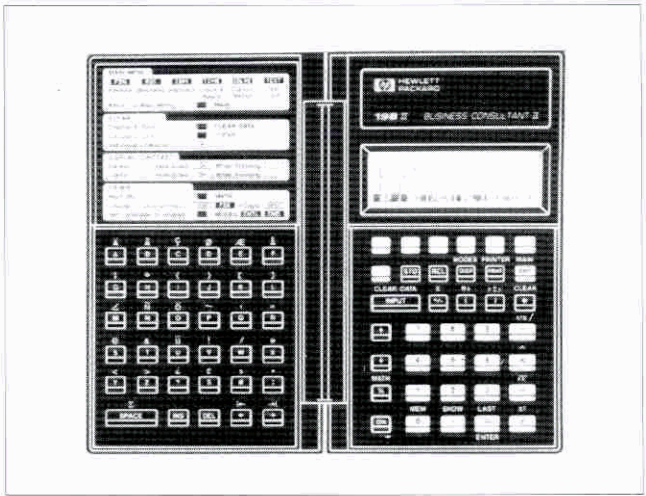
d) 1)))

Por último, introducimos m-1 y m+1 de forma contraria, y cerramos dos paréntesis. Uno es el situado inmediatamente después de Σ. El otro es el que se abre al calcular vt-1. Por eso xIF(FASE=:M+1:M-1)))

## RECTIFICACION DEL N.º 8
### MARZO 1993

Algunos lectores se han vuelto locos con la línea 14 del programa, ya que se ha escapado una errata tipográfica. No tiene la menor importancia, porque ya habéis comprobado que la longitud de las líneas coinciden con las que se visualizan en la pantalla. De todas formas, vamos a explicitarla un poco: En la línea 14 (pag. 58) falta un paréntesis y termina con un espacio en blanco. Atención en este último aspecto.

| | |
|---|---|
| Se escribe: | 1+I%÷100)^(T-1xIF(X>0 |
| debe ser: | 1+I%÷100)^(T-1)xIF(X>0 |



En pantalla queda:

```
INV  (1+I%x01)^(T)xIF(X>
0 AND Y>0:ITEM(M80:X+T)
xITEM(F80:Y+T):IF(X>0:I
TEM(M80:X+T):IF(Y>0:ITE
M(F80:Y+T):1)))xIF(FASE
=1:M-1:M+1)))
```

Si alguien no ha seguido todos los pasos, allá va de nuevo todo el programa con las modificaciones introducidas:

```
RENTA=IF(X>0 AND Y>0:I
NV(ITEM(M80:X)xITEM(F80
:Y)):IF(X>0:INV(ITEM(M8
0:X)):IF(Y>0:INV(ITEM(F
80:Y)):1)))xINV(2xM)xΣ(
T:1:IF(X>0 AND Y>0:MIN(
MIN(SIZES(M80)-X:SIZES(
F80)-Y):N):IF(X>0:MIN(S
IZES(M80)-X:N):IF(Y>O:M
IN   (SIZES(F80)-Y:N)N)))
:1:IF(FC=1:(1=c%x.01)^(
T-1):(1+C%x.01x(T-1)))x
(INV(1+I%x.01)^(T-1)xIF
(X>0 AND Y>0:ITEM(M80:X
+T-1)xITEM(F80:Y+T-1):I
F(X>0:ITEM(M80:X+T-1):I
F(Y>0:ITEM(F80:Y+T-1):1
)))xIF(FASE=1:M+1:M-1)+
INV(1+I%x,01)^(T)xIF(X>
0 AND Y>0:ITEM(M80:X+T)
xITEM(F80:Y+T):IF(X>0:I
TEM(M80:X+T):IF(Y>0:ITE
M(F80:Y+T):1)))xIF(FASE
=1:M-1:M+1)))
```

Un comentario más. Esta calculadora, en relación con otros instrumentos, posee poca memoria. Este es un dato importante, porque nos obliga a ser muy concisos. Por eso, habrá quien prefiera llamar a las tablas simplemente M y F en lugar de M80 y F80. Incluso, se podrían introducir varias tablas de mortalidad con nombres tales como P80, P81, R80, R81... y realizar el programa denominando a las tablas M y F. Sin embargo, en el momento de calcular deberemos elegir previamente dos tablas, a las que redenominaríamos con las letras M y F.

### III. Nuevos desarrollos. Conexiones

Este apartado queda abierto para todos aquellos que quieran seguir aportando ideas. No sería descabellado pensar incluso en la necesidad de abrir una sección informática dentro de la revista. Claro que eso depende de pequeñas aportaciones como las recogidas en estas páginas.

Para terminar el artículo y poder dejar abierto el tema únicamente voy a mencionar la posibilidad que existe de programar varias fórmulas, y conectar los valores de sus variables.

Para explicarlo, vamos a realizar un caso práctico. Supongamos que hemos programado la anterior fórmula, en la que x, la edad, es un dato. Sin embargo, lo que nosotros conocemos es la fecha de nacimiento y la de efecto.

En este caso, podemos hacer uso de una de las muchas funciones para desarrollar un programa en el que x sea la variable dependiente.

Por ejemplo, propongamos el siguiente:

X=RND(DDAYS(NAC:FTO:3)
+360:0)

Vamos a explicarlo, aunque se pueda entender muy fácilmente leyendo el manual.

La función DDAYS calcula el número de días entre dos fechas, a las que hemos denominado NAC y FTO. Nacimiento y fecha de efecto. El 3 significa que utiliza un calendario de 360 días, repartidos en 12 meses.

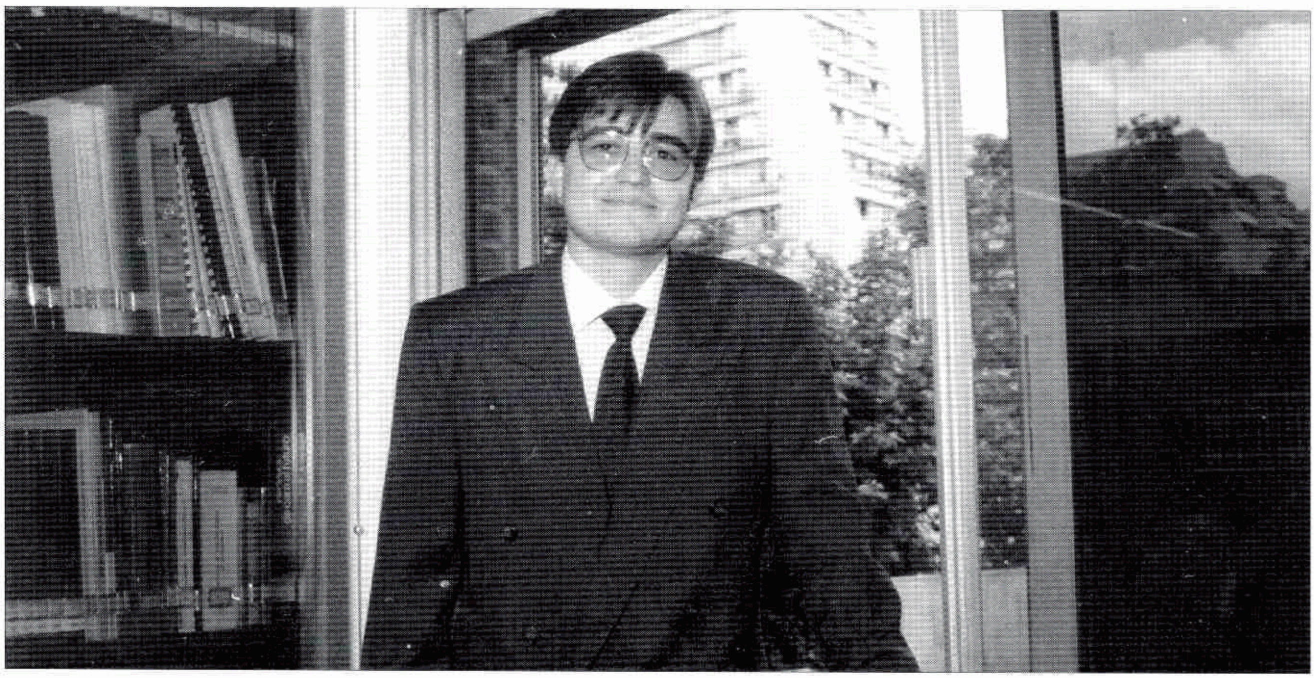
Este valor, si queremos hallar la edad actuarial con la aproximación descrita, habrá que redondearlo por exceso. Por eso usamos RND, que es otra función ya prevista en la programación no sólo en esta calculadora, sino en todos los lenguajes informáticos y hojas de cálculo. Para eliminar los decimales ponemos un 0.

Habrá que tener cuidado con el formato de la fecha. Esta puede venir como: DDMMAA, MMDDAA, MMDDAAAA...

Introducimos los valores deseados de NAC y FTO para hallar el valor de x.

En segundo lugar, presionamos (EXIT) y acudimos a la fórmula de la renta. El valor de x ya no habría que teclearlo. Por defecto, va a tomar el calculado anteriormente.

Interesante aplicación. ¿No es así? De nuevo, una vez expuesto cómo se puede programar, dejo en sus manos lo que quiera calcular... ■



*F. Javier Marcos, autor del artículo.*