

Escuela Politécnica Nacional

Metodos Numericos - Proyecto del I Bimestre

David Arciniegas, Darlin Anacicha, Angel Falcon, Joel Quilumba

2024-06-10

Tabla de contenidos

1	Objetivos	2
2	Introduccion	2
3	Metodologia	3
3.1	Descripción de la solución	3
3.2	Pasos para el uso del programa	4
3.2.1	Ejecución del programa	4
3.2.2	Ingreso de los datos	5
3.2.3	Calcular y mostrar los resultados	5
3.2.4	Visualización grafica de la evolución del capital	6
3.3	Formulas Matemáticas	6
3.4	Diagrama de flujo	8
3.4.1	Pseudocodigo	9
3.5	Detalles importantes de la implementación	11
4	Resultados	12
5	Conclusiones	13

1 Objetivos

- Crear una herramienta que permita a los usuarios calcular la evolución de sus ahorros a lo largo del tiempo, considerando un depósito inicial, aportes periódicos, frecuencias de aportes y una tasa de interés anual.
- Explicación de cómo el programa maneja aportes periódicos en distintas frecuencias y asegura la validez de los datos ingresados.
- Asegurar que los datos ingresados por el usuario sean apropiados, estableciendo límites mínimos para el depósito inicial y los aportes periódicos.
- Mostrar los resultados en una tabla que detalle cada periodo, incluyendo aportes, capital acumulado, intereses ganados y ganancia acumulada, así como un gráfico que visualice la evolución del capital y la ganancia acumulada a lo largo del tiempo.

2 Introduccion

Para este proyecto de ahorro semanal se desarrollado un programa el cual permite que el usuario calcule la evolución de sus ahorros a lo largo del tiempo basándose en un depósito inicial, aportes periódicos, frecuencias con que se realizan dichos aportes a sus ahorros y una tasa de interés anual.

Por otro lado, el programa gestiona aportes en distintas frecuencias, como semanal, mensual, bimestral y trimestral, lo que le permite adaptarse a diferentes necesidades y hábitos de ahorro de los usuarios. Además, se asegura de que los datos ingresados sean apropiados, estableciendo un depósito inicial mínimo de 50 dólares y aportes periódicos mínimos de 5 dólares.

Finalmente, el programa calcula el saldo final para cada periodo, incorporando tanto los aportes periódicos como los intereses acumulados. Utiliza la fórmula de interés compuesto para simular el crecimiento del capital, mostrando cómo las contribuciones regulares y el interés compuesto pueden incrementar significativamente los ahorros. Los resultados se presentan de forma clara en una tabla y un gráfico.

3 Metodologia

3.1 Descripción de la solución

La primera etapa del programa es la creación de la función “calcular_ahorro”, que requiere el depósito inicial, los aportes periódicos, la frecuencia de los aportes, la tasa de interés anual y el número de periodos como parámetros. Seguido calcula la tasa de interés periódica dividiendo la tasa anual por el número de periodos por año. Luego, la función recorre cada periodo y suma los aportes y calcula el interés compuesto para cada periodo.

Luego tenemos la función “graficar_resultados” que nos mostrara el incremento que ha tenido el capital según los aportes que se haya realizado dependiendo el tiempo del periodo, además podemos actualizar cualquiera de los parámetros para probar múltiples escenarios y ver instantáneamente cómo diferentes las configuraciones de depósito inicial, frecuencia de aportes y tasas de interés afectan el crecimiento del ahorro.

Seguido tenemos la creación de la interfaz grafica con que la interactúa el usuario para ingresar cada uno de los datos una vez realizado eso damos un click en “calcular_y_mostrar” y se nos desglosa en la tabla todos cálculos realizados. En caso de que el usuario ingresa datos no válidos, como un depósito inicial menor a 50 dólares o aportes periódicos menores a 5 dólares, se mostrara un mensaje de error.

3.2 Pasos para el uso del programa

3.2.1 Ejecución del programa

Ejecución del código para acceder a la interfaz gráfica

Calculadora de Ahorro

Depósito Inicial (\$):

Aporte Periódico (\$):

Frecuencia:

Cantidad de Años:

Periodo	Aporte (\$)	Capital (\$)	Ganancia (\$)	Ganancia Acumulada (\$)	Total (\$)
---------	-------------	--------------	---------------	-------------------------	------------

3.2.2 Ingreso de los datos

Ingreso de cada uno de los valores en cada campo.

Calculadora de Ahorro

Depósito Inicial (\$): 100

Aporte Periódico (\$): 5

Frecuencia: semanal

Cantidad de Años: 2

Calcular y Mostrar

Periodo	Aporte (\$)	Capital (\$)	Ganancia (\$)	Ganancia Acumulada (\$)	Total (\$)
---------	-------------	--------------	---------------	-------------------------	------------

3.2.3 Calcular y mostrar los resultados

Visualización de los cálculos correspondientes en la tabla

Calculadora de Ahorro

Depósito Inicial (\$): 100

Aporte Periódico (\$): 5

Frecuencia: semanal

Cantidad de Años: 4

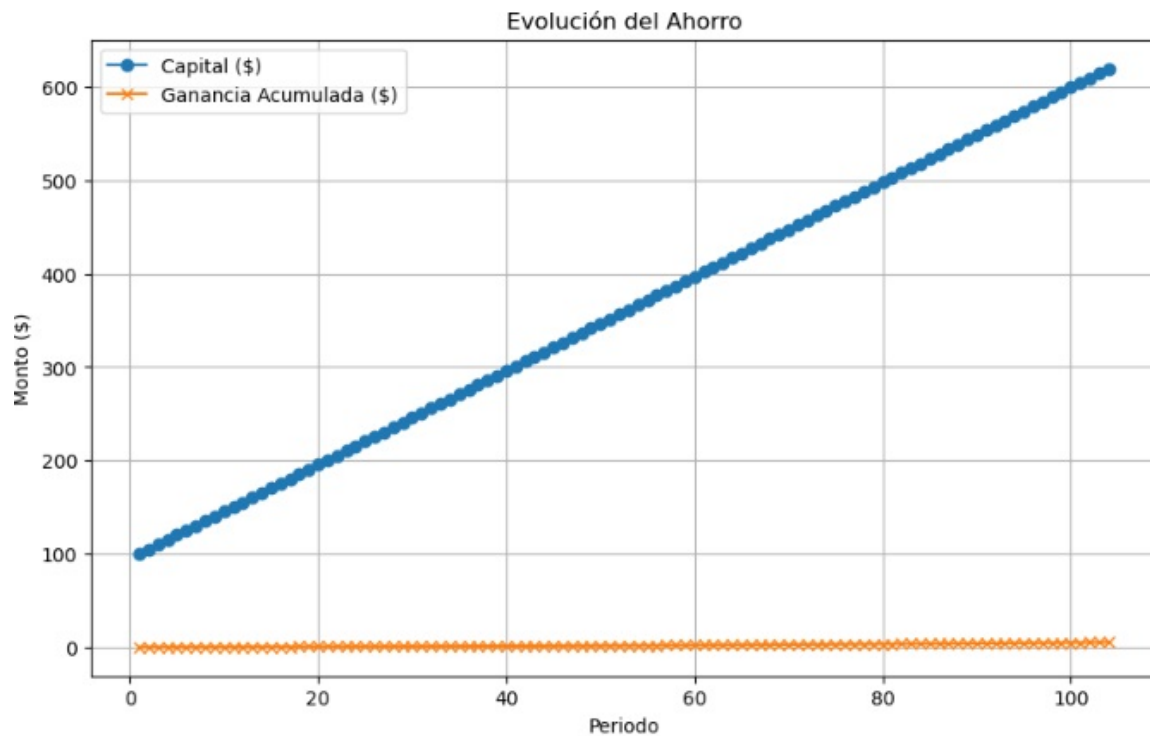
Calcular y Mostrar

Periodo	Aporte (\$)	Capital (\$)	Ganancia (\$)	Ganancia Acumulada (\$)	Total (\$)
1	100.0	100.01	0.01	0.01	100.01
2	5.0	105.03	0.01	0.03	105.03
3	5.0	110.04	0.01	0.04	110.04
4	5.0	115.06	0.02	0.06	115.06
5	5.0	120.07	0.02	0.07	120.07
6	5.0	125.09	0.02	0.09	125.09
7	5.0	130.11	0.02	0.11	130.11
8	5.0	135.13	0.02	0.13	135.13
9	5.0	140.14	0.02	0.14	140.14
10	5.0	145.16	0.02	0.16	145.16

Frecuencia: semanal
Periodos por Año: 52
Tasa Periodica Inicial: 0.05000
Tasa Periodica Optima: 0.01338
Tasa Anual Calculada: 0.70%

3.2.4 Visualización grafica de la evolución del capital

Gráfico que ilustra la evolución del capital y del total acumulado a lo largo de los periodos.



3.3 Formulas Matemáticas

Metodo de Newthton-Raphson

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

De donde:

- x_n es la aproximacion actual
- $f(x_n)$ es la funcion de la cual queremos encontrar la raiz
- $f'(x_n)$ es la derivada de $f(x_n)$

Ecuacion de la tasa periodica

$$F(r_p) = P * (1 + \frac{r_p}{100})^m + A * (\frac{(1 + \frac{r_p}{100})^m - 1}{\frac{r_p}{100}}) - (P + A * m)$$

De donde:

- P : Depósito inicial.
- A : Aporte periódico.
- n : Número de periodos en un año.
- m : Total de periodos (años x periodos por año).
- r_p : Tasa de interés periódica (en %).

El metodo de Newton-Raphson ajusta r_p hasta que $F(r_p) \approx 0$

Ejemplo

Supongamos que $P=1000$, $A=100$, $n=12$, $m=60$ y que el calculo de Newthron-Raphson encuentra $r_p \approx 0.04746\%$

Tasa periodica optima encontrada = 0.04746%

Conversion a tasa anual =

$$r_a = r_p * n = 0.04746 * 12 \approx 0.57\%$$

3.4 Diagrama de flujo



3.4.1 Pseudocodigo

Algoritmo Ahorro_Semanal

Definir ventana Como Objeto

Definir entry_deposito_inicial, entry_aporte_periodico, combo_frecuencia, entry_tasa_anual,

SubProceso calcular_ahorro (deposito_inicial, aporte_periodico, frecuencia, tasa_anual, num_periodos)

Definir frecuencia_dict Como Diccionario

frecuencia_dict["semanal"] = 52

frecuencia_dict["mensual"] = 12

frecuencia_dict["bimestral"] = 6

frecuencia_dict["trimestral"] = 4

Si frecuencia No Esta En frecuencia_dict Entonces

Mostrar Mensaje "Frecuencia no válida. Use 'semanal', 'mensual', 'bimestral' o 'trimestral'"

Fin SubProceso

Fin Si

Definir periodos_por_ano, tasa_periodica Como Real

periodos_por_ano = frecuencia_dict[frecuencia]

tasa_periodica = tasa_anual / periodos_por_ano

Definir capital, interes Como Real

capital = deposito_inicial

Definir resultados Como Matriz

Para periodo = 1 Hasta num_periodos Con Paso 1 Hacer

Si periodo > 1 Entonces

capital = capital + aporte_periodico

Fin Si

interes = capital * (tasa_periodica / 100)

capital = capital + interes

Agregar Elemento A resultados: (periodo, Si periodo > 1 Entonces

aporte_periodico SiNo deposito_inicial,

Redondear(capital, 2), Redondear(interres, 2), Redondear(capital, 2))

Fin Para

retornar resultados

Fin SubProceso

SubProceso graficar_resultados (resultados Como Matriz)

Definir df Como Objeto

```

df <- Crear DataFrame Con resultados

Crear Figura de Tamaño (10, 6)
Plotear df["Periodo"], df["Capital ($)"], marcador 'o', etiqueta 'Capital ($)'
Plotear df["Periodo"], df["Total ($)"], marcador 'x', etiqueta 'Total ($)'
Etiquetar Eje X 'Periodo'
Etiquetar Eje Y 'Monto ($)'
Título 'Evolución del Ahorro'
Mostrar Leyenda
Mostrar Grilla
Mostrar Gráfica
Fin SubProceso

```

```

SubProceso calcular_y_mostrar
  Definir deposito_inicial, aporte_periodico, tasa_anual Como Real
  Definir frecuencia Como Cadena
  Definir num_periodos Como Entero

  deposito_inicial = Convertir A Real(entry_deposito_inicial)
  aporte_periodico = Convertir A Real(entry_aporte_periodico)
  frecuencia = combo_frecuencia
  tasa_anual = Convertir A Real(entry_tasa_anual)
  num_periodos = Convertir A Entero(entry_num_periodos)

  Si deposito_inicial < 50 Entonces
    Mostrar Mensaje "El depósito inicial debe ser al menos 50 dólares."
  Fin SubProceso
Fin Si

  Si aporte_periodico < 5 Entonces
    Mostrar Mensaje "El aporte periódico debe ser al menos 5 dólares."
  Fin SubProceso
Fin Si

  Definir resultados Como Matriz
  resultados = calcular_ahorro(deposito_inicial,
    aporte_periodico, frecuencia, tasa_anual, num_periodos)

  Limpiar Tabla tree

  Para Cada fila En resultados Hacer
    Insertar fila En tree
  Fin Para

```

```

        graficar_resultados(resultados)
Fin SubProceso

// Crear la ventana principal
ventana <- CrearVentana("Calculadora de Ahorro")

// Crear y ubicar los widgets
CrearLabel(ventana, "Depósito Inicial ($) :", 0, 0)
entry_deposito_inicial <- CrearEntrada(ventana, 0, 1)

CrearLabel(ventana, "Aporte Periódico ($) :", 1, 0)
entry_aporte_periodico <- CrearEntrada(ventana, 1, 1)

CrearLabel(ventana, "Frecuencia :", 2, 0)
combo_frecuencia <- CrearCombo(ventana, ["semanal", "mensual", "bimestral", "trimestral"])

CrearLabel(ventana, "Tasa de Interés Anual (%) :", 3, 0)
entry_tasa_anual <- CrearEntrada(ventana, 3, 1)

CrearLabel(ventana, "Número de Periodos :", 4, 0)
entry_num_periodos <- CrearEntrada(ventana, 4, 1)

CrearBoton(ventana, "Calcular y Mostrar", calcular_y_mostrar, 5, 0, 2)

// Crear la tabla para mostrar los resultados
columns <- ["Periodo", "Aporte ($)", "Capital ($)", "Ganancia ($)", "Total ($)"]
tree <- CrearTabla(ventana, columns, 6, 0, 2)

EjecutarVentana(ventana)
Fin Algoritmo

```

3.5 Detalles importantes de la implementación

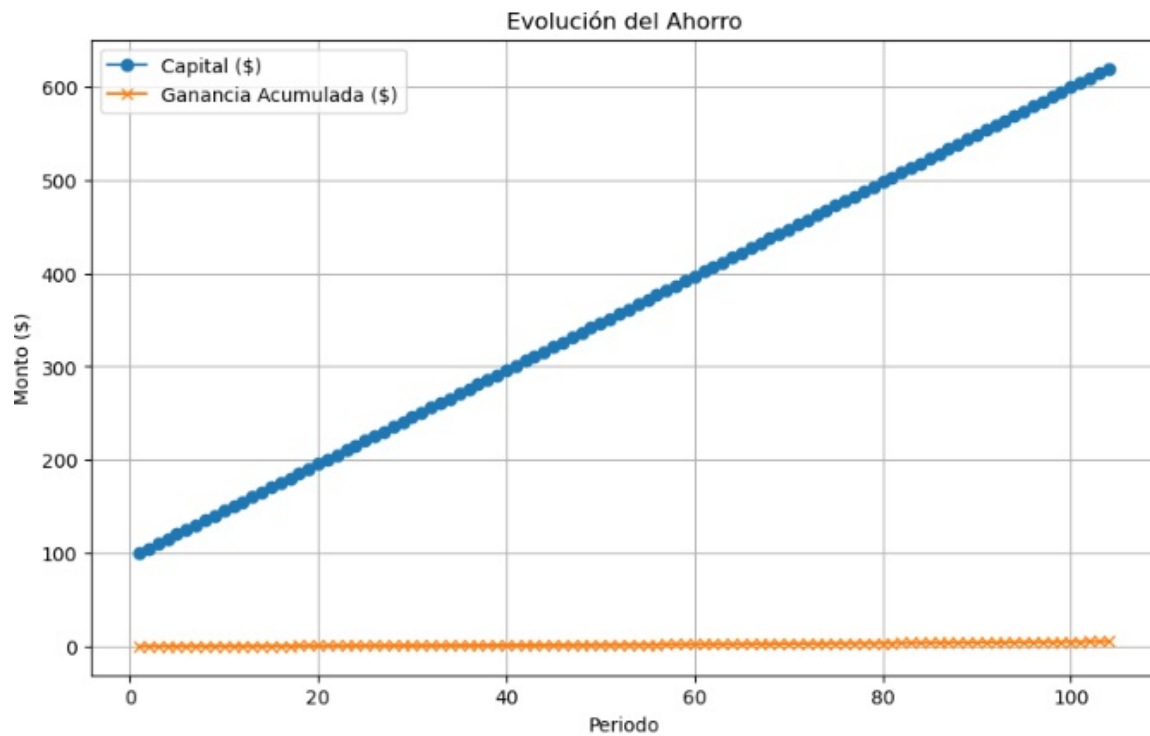
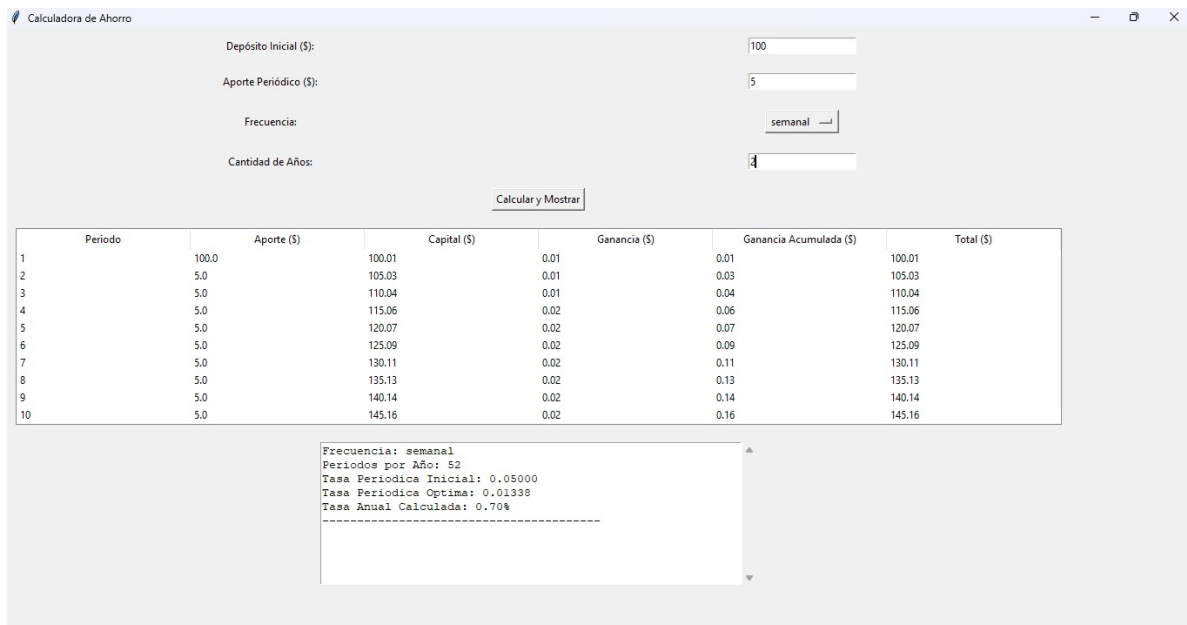
Interfaz de Usuario: Se emplea la biblioteca tkinter para diseñar una interfaz gráfica que permite al usuario ingresar los datos necesarios y visualizar los resultados.

Cálculo del Ahorro La función `calcular_ahorro` lleva a cabo los cálculos necesarios para determinar cómo evoluciona el ahorro.

Gráfica de Resultados: La función `graficar_resultados` genera una gráfica con matplotlib que muestra la evolución del capital y el total acumulado.

Validación de Datos: Se aplica una validación básica para garantizar que los valores de entrada sean apropiados (depósito inicial y aporte periódico mínimos).

4 Resultados



5 Conclusiones

1. El aspecto mas importante del proyecto fue encontrar la función para calcular el interés.
2. El programa desarrollado ofrece una gran flexibilidad y adaptabilidad a las necesidades de los usuarios al permitirles seleccionar diferentes frecuencias de aportes (semanal, mensual, bimestral y trimestral).
3. El programa desarrollado ofrece una gran flexibilidad y adaptabilidad a las necesidades de los usuarios al permitirles seleccionar diferentes frecuencias de aportes (semanal, mensual, bimestral y trimestral).
4. El programa implementa mecanismos de validación para asegurar que los datos ingresados por los usuarios sean apropiados, estableciendo límites mínimos para el depósito inicial (50 dólares) y los aportes periódicos (5 dólares)
5. La función de actualizar parámetros y recalculando instantáneamente los resultados permite a los usuarios explorar múltiples escenarios de ahorro.