

**Aplicativo análisis de sensibilidad**

Manual de Usuario

Versión: 0100

Fecha: 03/02/2021

**HOJA DE CONTROL**

|  |  |  |  |
| --- | --- | --- | --- |
| **Organismo** | Investigación de Operaciones I - Grupo 2 | | |
| **Proyecto** | Aplicativo análisis de sensibilidad financiero | | |
| **Entregable** | Manual de Usuario | | |
| **Autores** | Diego González - 20171020111  Esteban Olaya - 20171020135  Jairo Buitrago - 201701020002 | | |
| **Versión/Edición** | 0100 | **Fecha Versión** | 03/02/2021 |
|  |  | **Nº Total de Páginas** |  |

REGISTRO DE CAMBIOS

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Causa del Cambio** | **Responsable del Cambio** | **Fecha del Cambio** |
| 0100 | Versión inicial | Diego González - 20171020111  Esteban Olaya - 20171020135  Jairo Buitrago - 201701020002 | 03/02/2021 |

**ÍNDICE**

[**DESCRIPCIÓN DEL SISTEMA**](#_1fob9te) **3**

[Objetivo](#_3znysh7) 4

[Objetivos de la aplicación](#_mv9o5cu94ap) 4

[Alcance](#_2et92p0) 4

[Funcionalidad](#_tyjcwt) 4

[**MAPA DEL SISTEMA**](#_3dy6vkm) **4**

[Modelo Lógico](#_1t3h5sf) 5

[**DESCRIPCIÓN DEL SISTEMA**](#_ksfajrdf3bmc) **6**

[Implementación](#_17dp8vu) 8

[Mensajes de error](#_26in1rg) 10

[Ayudas contextuales](#_lnxbz9) 11

[**GLOSARIO**](#_35nkun2) **13**

[**BIBLIOGRAFÍA Y REFERENCIAS**](#_2jxsxqh) **14**

[**Repositorio**](#_hyxtnfbk0znd) **14**

# DESCRIPCIÓN DEL SISTEMA

## Objetivo

El objetivo de este documento, es dar a conocer a los usuarios finales todas las características del sistema, como también la forma adecuada de utilizarlo. Todo esto mediante la explicación detallada e ilustrada de las funcionalidades y cómo interactúan en el sistema. Además de encontrar una sección de preguntas frecuentes y un glosario con términos que se pueden considerar confusos o poco claros.

## Objetivos de la aplicación

* Crear un aplicativo que nos permita hacer análisis de sensibilidad financieros.
* Encontrar una solución que ayude a las personas a orientarse sobre la capacidad de pago que puedan tener en sus préstamos y a cuánto plazo pueden hacerlo.

## Alcance

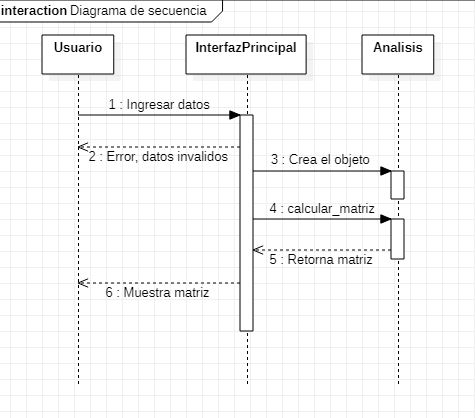
Aplica a todos los usuarios que deseen hacer uso de la aplicación para orientarse en cuanto a la capacidad de pago que puedan tener sus préstamos y el plazo en el cual lo puedan realizar, al igual que aquellos interesados en el estudio de los componentes y desarrollo de la aplicación.

## Funcionalidad

Este aplicativo permite realizar análisis de sensibilidad financieros sobre préstamos o créditos que se puedan solicitar en las entidades financieras, recibiendo como parámetros la cantidad de cuotas que está dispuesto a pagar el deudor, el valor de la cuota y la tasa de interés, esta funcionalidad puede ser útil para cualquier persona que requiera financiamiento en un proyecto de cualquier índole, ya sea de carácter social o privado, cualquier emprendimiento o simplemente para una persona que quiera adquirir cualquier bien o servicio. La aplicación le permitirá conocer el monto final que podría prestarle la entidad financiera y si su poder adquisitivo le permitirá realizar el pago de las cuotas oportunamente.

# MAPA DEL SISTEMA

## Modelo Lógico



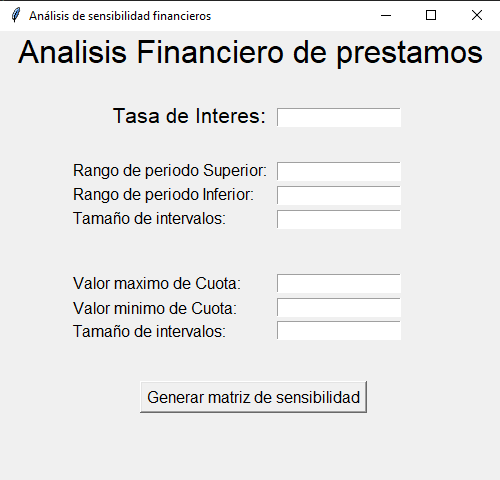
Para representar el mapa lógico del sistema, utilizamos un diagrama de secuencia el cual nos guía a través de los componentes del sistema y cómo estos interactúan.

La primer acción que se realiza es el ingreso de los datos por parte del usuario, el cual interactúa con nuestro componente de la interfaz gráfica que tiene como nombre “InterfazPrincipal”. Este componente realiza una verificación de los datos que en caso de ser incorrectos, devolverá un mensaje de error informando al usuario qué ha sucedido y en caso de que el usuario ingrese de los datos de forma adecuada, el componente de la interfaz principal procede a instanciar un objeto de la clase “Analisis”, el cual recibirá unos parámetros que serán explicados más adelante.

Ya instanciado el objeto, el componente de la interfaz gráfica, hace un llamado al método “calcular\_matriz” el cual contiene el llamado a otros métodos que permiten el cálculo de la matriz que finalmente es devuelta al componente “InterfazPrincipal”, quien a su vez lo muestra al usuario, dando fin a la ejecución del sistema.

# DESCRIPCIÓN DEL SISTEMA

Al ejecutar el aplicativo lo primero que encontramos es una pantalla que contiene 7 campos, el primero será un espacio para que el usuario ingrese la tasa de interés ofrecida por la entidad de la cual obtendrá el financiamiento o crédito, los siguientes tres serán los espacios para el ingreso de los valores extremos del periodo y el tamaño que determina los intervalos a analizar y por último los 3 campos finales serán para los valores de las cuotas, al igual que con los periodos serán los valores máximo, mínimo y el tamaño del intervalo a analizar.



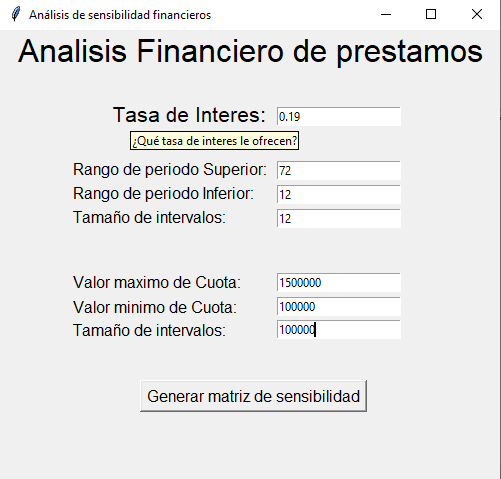
Con el objetivo de apoyar al usuario en el uso de la aplicación, hemos colocado mensajes emergentes en todos los campos de la aplicación, el usuario podrá visualizarlos al arrastrar el cursor sobre los textos a los laterales:



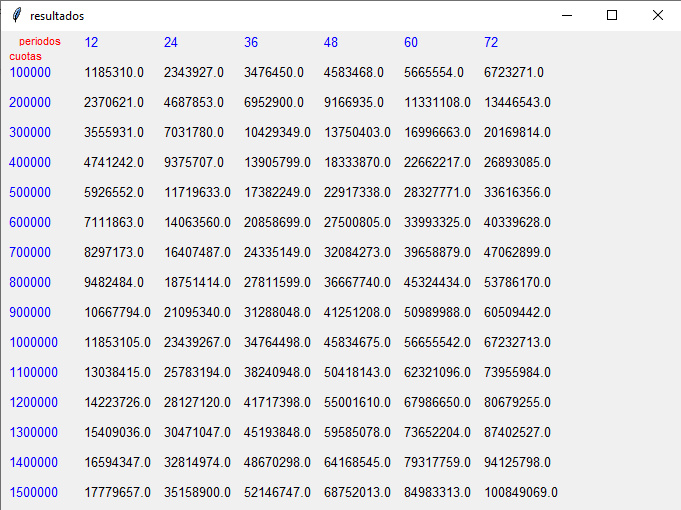




Rellenaremos los campos con los valores que el usuario desee, para este ejemplo utilizaremos los siguientes:



Luego de llenar los campos con nuestros valores le daremos al botón “Generar matriz de sensibilidad”, lo cual nos arrojará la siguiente matriz:



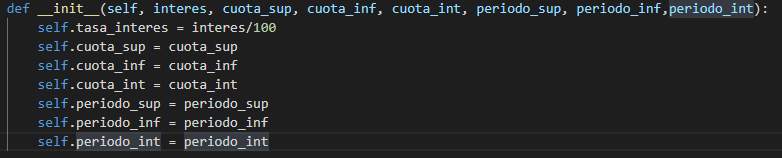
Lo que veremos en la matriz serán los posibles valores de financiamiento dado un número de periodos y un valor de las cuotas, por lo cual el usuario podrá tomar una decisión que le permita obtener el crédito que cubra sus necesidades y que se encuentre dentro de sus capacidades de pago.

## Implementación

Para la implementación del aplicativo se diseñó una clase que nos permitiera hacer los diferentes cálculos requeridos para el análisis de sensibilidad, a esta clase se le llamó “Analisis”.



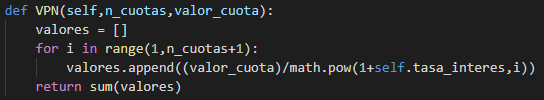
Posterior a la definición de la clase, se implementó un constructor que nos permitiera inicializar los diferentes atributos de clase.



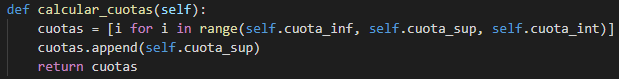
Como se puede notar se definieron varios atributos. los cuales se describirán a continuación:

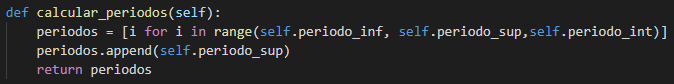
* “tasa\_interes” → Interés del financiamiento o crédito que se desea obtener.
* “cuota\_sup” → Rango superior o máximo a pagar por el valor de una cuota.
* “cuota\_inf” → Rango inferior o mínimo a pagar por el valor de una cuota.
* “cuota\_int” → Tamaño de los intervalos a analizar dentro de la matriz.
* “periodo\_sup” → Tiempo máximo a tener en cuenta para el pago de las cuotas.
* “periodo\_inf” → tiempo mínimo a tener en cuenta para el pago de las cuotas.
* “periodo\_int” → Intervalos de tiempo para el análisis.

Declaración de la función “VPN” que se encargará de calcular el valor posible de un financiamiento dado el número de cuotas, el costo de las mismas y la tasa de interés.

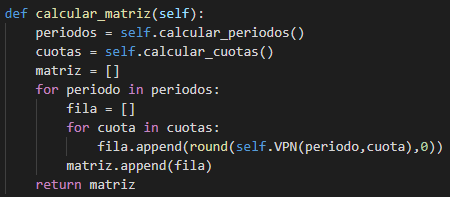


Funciones que retornan listas, las cuales contienen los diferentes valores de las cuotas, dado su rango minimo, rango maximo y su tamaño de intervalo, así mismo se realiza con los periodos.



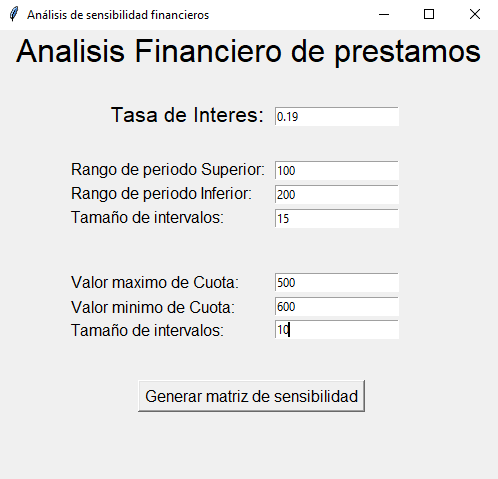


Método que calcula la matriz de sensibilidad, primero generando las listas de periodos y cuotas declaradas en los métodos anteriores y utilizando la función “VPN” para calcular el valor del financiamiento, todas las filas que genera las agrega a la matriz y finalmente la retorna.

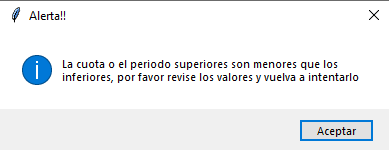


### Mensajes de error

En el programa tenemos una situación donde se podrían ingresar mal los datos referentes a la generación de la matriz y por tanto el programa podría fallar, por ello hemos decidido añadir un mensaje de alerta que indique al usuario dónde está cometiendo el error, el problema se presenta cuando un usuario añade valores más grandes para los rangos inferiores del periodo o las cuotas que los valores que ingresa en el rango superior. En la siguiente imagen se puede visualizar el ejemplo de unos datos más ingresados:



Para que el programa tenga respuesta a este tipo de casos hemos agregado un mensaje de alerta como el que se muestra a continuación:



### Ayudas contextuales

Para entender mejor cómo fue elaborado ese sistema, adjuntamos unas ayudas contextuales que servirán de guía en caso de no entender algún procedimiento mencionado anteriormente al igual que mostrar algunas de las herramientas utilizadas para la realización de este producto de software.

**Valor presente Neto(VPN)**: Seguramente viste estas siglas anteriormente, estas corresponden a lo que se conoce como valor presente neto.  
El valor presente neto es el valor de los flujos de efectivo proyectados, descontados al presente. Es un método de modelado financiero utilizado por los contadores para la elaboración de presupuestos de capital y por analistas e inversores para evaluar la rentabilidad de las inversiones y proyectos propuestos.

**Método del valor presente neto**

El método del valor presente neto incorpora el valor del dinero en determinado tiempo de flujos de efectivo netos de un negocio o proyecto. El objetivo del valor presente neto es realizar las comparaciones entre los periodos en donde el proyecto o negocio tuvo diferentes flujos de efectivo para determinar si conviene o no invertir en él.

El valor depende de la tasa de interés a la que se ajuste el cálculo del valor presente neto. El método para calcular el valor presente neto es a través de un quebrado o razón aritmética en donde el numerador de dicha razón es dinero y el denominador es la tasa de interés.

**Fórmula:**

Para calcular el valor presente neto (VPN) hay que visualizar todos los flujos de caja hasta el momento actual y descontarles el tipo de interés determinado. De esta forma, cada entrada y salida de efectivo se descontará del valor actual y se sumará, tal como lo establece la siguiente fórmula.

**Librerías y paquetes:** Para el desarrollo de este aplicativo se hizo uso de varias librerías y paquetes que nos proporciona el lenguaje de programación python, dentro de ellas se encuentra: NumPy, math, matplotlib y tkinter. Unas de ellas utilizadas para las operaciones y todo lo referente al manejo de datos, mientras que las otras nos brindan la facilidad de construir una interfaz gráfica, para hacer más amigable el uso del aplicativo.

A continuación describiremos brevemente cada una de las librerías y paquetes:

**NumPy**

Es una biblioteca para el lenguaje de programación Python que da soporte para crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas.

**Math**

Es un módulo que está incluido en la librería estándar de Python, el cual provee operaciones matemáticas como suma, resta, multiplicación y división, además de una serie de funciones más avanzadas como lo son. Funciones trigonométricas y logarítmicas, valor absoluto, cálculo de la distancia euclídea, potenciación y radicación, entre muchas otras.

**Tkinter**

Es un binding de la biblioteca gráfica Tcl/Tk para el lenguaje de programación Python. Se considera un estándar para la interfaz gráfica de usuario para Python y es el que viene por defecto con la instalación para Microsoft Windows.

**Matplotlib**

Es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy.

Si desea conocer más acerca de estos paquetes y librerías, en la bibliografía encuentra el enlace hacia la documentación oficial de cada herramienta.

# GLOSARIO

|  |  |
| --- | --- |
| **Término** | **Descripción** |
| Python | Es un lenguaje de programación interpretado y multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. |
| Librería | Es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca. |
| Interfaz gráfica | También conocida también como GUI, es una vista que utilizando un conjunto de imágenes y objetos gráficos representan la información y acciones disponibles en nuestro sistema. |
| Clase | Es una plantilla o molde para la creación de objetos de datos según un modelo predefinido. Las clases se utilizan para representar entidades o conceptos, como los sustantivos en el lenguaje |
| Instancia | Es la particularización, realización específica u ocurrencia de una determinada clase, entidad o prototipo. En los lenguajes de programación orientada a objetos un objeto es una instancia de una clase. |
| Método | Método es una subrutina cuyo código es definido en una clase y puede pertenecer tanto a una clase, como es el caso de los métodos de clase o estáticos, como a un objeto, como es el caso de los métodos de instancia. |

# BIBLIOGRAFÍA Y REFERENCIAS

* Gasbarrino, Stefano. (25 de noviembre de 2020). Valor Presente Neto: qué es y cómo se calcula (incluye ejemplo) [Mensaje en un blog]. Recuperado de <https://blog.hubspot.es/sales/que-es-valor-presente-neto>
* NumPy Documentation. NumPy. Recuperado de <https://numpy.org/doc/>
* The Matplotlib development team. (12 de noviembre de 2020). Overview. Matplotlib. Recuperado de <https://matplotlib.org/contents.html>.
* (22 de enero de 2021). math — Mathematical functions. Python. Recuperado de <https://docs.python.org/3/library/math.html>.
* (22 de enero de 2021). tkinter — Python interface to Tcl/Tk. Python. <https://docs.python.org/3/library/tkinter.html>.}
* Análisis de sensibilidad de crédito. Análisis realizado con hojas de excel. [Análisis de Sensibilidad (finanzas9y6.com)](https://www.finanzas9y6.com/analisis-de-sensibilidad/)
* [Tasas de interés de política monetaria | Banco de la República (banco central de Colombia) (banrep.gov.co)](https://www.banrep.gov.co/es/estadisticas/tasas-interes-politica-monetaria)

# 6 Repositorio

[IO1/Proyecto at main · jairosam/IO1 (github.com)](https://github.com/jairosam/IO1/tree/main/Proyecto)