MÉTODOS NUMÉRICOS

Y DESARROLLO DE HERRAMIENTAS PARA SU SOLUCIÓN

NetBeans

Proyectos desarrollados en colaboración con los alumnos del curso de Métodos Numéricos de la E.P. de Ingeniería Industrial - UNHEVAL

Guillermo Augusto Bocangel Weydert
Elmer Santiago Chuquiyauri Saldivar
Guillermo Augusto Bocangel Marin
Nérida del Carmen Pastrana Díaz
Jorge Rubén Hilario Cárdenas
Jhonny Henry Piñán García
Hernan Wilmer Garcia Bonilla
Lincol Jarly Gomez Meza

Atena Editora
Ano 2024

Editora jefe
Profª Drª Antonella Carvalho de Oliveira
Editora ejecutiva
Natalia Oliveira
Asistente editorial
Flávia Roberta Barão
Bibliotecario
Janaina Ramos

Métodos numéricos y desarrollo de herramientas para su solución

Indexación: Amanda Kelly da Costa Veiga
Revisión: Los autores

Datos de catalogación en publicación internacional (CIP)

M593 Métodos numéricos y desarrollo de herramientas para su solución / Guillermo Augusto Bocangel Weydert, Elmer Santiago Chuquiyauri Saldivar, Guillermo Augusto Bocangel Marin, et al. – Ponta Grossa - PR: Atena, 2024.

Otros autores
Nérida del Carmen Pastrana Díaz
Jorge Rubén Hilario Cárdenas
Jhonny Henry Piñán García
Hernan Wilmer Garcia Bonilla
Lincol Jarly Gomez Meza
NetBeans

Formato: PDF
Requisitos del sistema: Adobe Acrobat Reader
Modo de acceso: World Wide Web
Incluye bibliografia
ISBN 978-65-258-2984-5
DOI: https://doi.org/10.22533/at.ed.845242611

1. Engenharia industrial. I. Weydert, Guillermo Augusto Bocangel. II. Saldivar, Elmer Santiago Chuquiyauri. III. Marin, Guillermo Augusto Bocangel. IV. Título.
CDD 670.427

Preparado por Bibliotecario Janaina Ramos – CRB-8/9166

Atena Editora
Ponta Grossa – Paraná – Brasil
Telefone: +55 (42) 3323-5493
www.atenaeditora.com.br
contato@atenaeditora.com.br

DECLARACIÓN DEL AUTOR

Para efectos de esta declaración, el término 'autor' se utilizará de forma neutral, sin distinción de género o número, salvo que se indique lo contrario. De esta misma forma, el término 'obra' se refiere a cualquier versión o formato de creación literaria, incluidos, pero no limitando a artículos, e-books, contenidos en línea, de acceso abierto, impresos y/o comercializados, independientemente del número de títulos o volúmenes. El autor de esta obra: 1. Atestigua que no tiene ningún interés comercial que constituya un conflicto de intereses en relación con la obra publicada; 2. Declara que participó activamente en la elaboración de la obra, preferentemente en: : a) Concepción del estudio, y/o adquisición de datos, y/o análisis e interpretación de datos; b) Preparación del artículo o revisión con el fin de que el material sea intelectualmente relevante; c) Aprobación final de la obra para su presentación; 3. Certifica que la obra publicada está completamente libre de datos y/o resultados fraudulentos; 4. Confirma la citación y referencia correcta de todos los datos e interpretaciones de datos de otras investigaciones; 5. Reconoce haber informado todas las fuentes de financiamiento recibidas para realizar la investigación; 6. Autoriza la edición de la obra, que incluye registros de la ficha catalográfica, ISBN, DOI y otros indexadores, diseño visual y creación de portada, maquetación del núcleo, así como su lanzamiento y difusión según los criterios de Atena Editora.

DECLARACIÓN DE LA EDITORIAL

Atena Editora declara, para todos los efectos legales, que: 1. La presente publicación sólo constituye una cesión temporal de los derechos de autor, del derecho de publicación, y no constituye responsabilidad solidaria en la creación de la obra publicada, en los términos de la Ley de Derechos de Autor (Ley 9610/98), del art. 184 del Código Penal y del art. 927 del Código Civil; 2. Autoriza e incentiva a los autores a firmar contratos con repositorios institucionales, con el fin exclusivo de divulgar la obra, siempre que se reconozca debidamente la autoría y edición y sin ningún fin comercial; 3. La editorial puede poner la obra a disposición en su sitio web o aplicación, y el autor también puede hacerlo a través de sus propios medios. Este derecho solo se aplica en caso de que la obra no se comercialice a través de librerías, distribuidores o plataformas asociadas. Cuando la obra se comercialice, los derechos de autor se cederán al autor al 30% del precio de cubierta de cada ejemplar vendido; 4. Todos los miembros del consejo editorial son doctores y están vinculados a instituciones públicas de educación superior, conforme a lo recomendado por CAPES para la obtención del libro Qualis; 5. De conformidad con la Ley General de Protección de Datos (LGPD), la editorial no cede, comercializa o autoriza el uso de los nombres y correos electrónicos de los autores, ni ningún otro dato sobre los mismos, para cualquier finalidad que no sea la divulgación de esta obra.

INDICE

Presentación del Libro

"Métodos Numéricos y Desarrollo de Herramientas para su Solución" es una contribución significativa al campo de la ingeniería y las ciencias aplicadas, realizada por los ingenieros: Elmer Chuquiyauri Saldivar, Guillermo Augusto Bocangel Weydert, Jhonny Henry Piñán García, Guillermo Augusto Bocangel Marín, Jorge Rubén Hilario Cárdenas, Hernan Wilmer Garcia Bonilla y Lincol Jarly Gomez Meza. Esta materia de ayuda es fruto de una colaboración intensiva entre los autores y los estudiantes del curso de Métodos Numéricos de la Escuela Académico Profesional de Ingeniería Industrial de la Universidad Nacional Hermilio Valdizán (UNHEVAL), que ha resultado en una amalgama de teoría robusta y aplicaciones prácticas de la matemática numérica en la resolución de problemas complejos de ingeniería.

El libro se destaca por su enfoque práctico, ofreciendo no solo un sólido fundamento teórico sino también guías paso a paso para la implementación de métodos numéricos en software contemporáneo como MATLAB y Java NetBeans. La obra está diseñada para ser una herramienta esencial tanto para estudiantes que se inician en el campo como para profesionales que buscan profundizar y actualizar sus conocimientos técnicos.

La estructura del libro facilita la comprensión al combinar teoría detallada con ejemplos aplicados y estudios de caso que demuestran la relevancia de los métodos numéricos en aplicaciones reales. Cada capítulo introduce métodos específicos, discutiendo sus fundamentos, la lógica detrás de sus fórmulas, y su ejecución en programas de computadora, permitiendo al lector seguir el proceso de transformar un problema matemático abstracto en una solución computable y práctica.

Este enfoque integrador no solo refuerza la comprensión teórica sino que también mejora las habilidades prácticas de los lectores, preparándolos para enfrentar desafíos reales en sus campos profesionales. Los autores han hecho un esfuerzo consciente para que el libro sirva como un recurso duradero en la biblioteca de cualquier ingeniero, científico o matemático aplicado, proporcionando un recurso valioso para la enseñanza, la práctica profesional o incluso como guía de consulta.

Introducción

Los métodos numéricos son esenciales para el científico e ingeniero moderno, proporcionando herramientas para resolver ecuaciones que son insuperables por métodos analíticos tradicionales. Este libro comienza estableciendo una comprensión sólida de la teoría de errores, fundamental para realizar cálculos numéricos eficientes y precisos, y progresivamente guía al lector a través de técnicas más complejas y sus aplicaciones.

El primer capítulo no solo establece una base teórica sobre errores en cálculos numéricos, sino que también prepara el terreno para los capítulos subsiguientes que cubren métodos específicos como la bisección, Newton-Raphson, y otros métodos iterativos y su aplicación usando software especializado. Este enfoque metódico no solo demuestra cómo se implementan estos métodos en la práctica, sino que también discute sus limitaciones y cómo superarlas en situaciones prácticas.

Asimismo, cada capítulo incluye ejercicios prácticos diseñados para consolidar la comprensión de los lectores y su capacidad para aplicar estos métodos a problemas reales. Estos problemas son cuidadosamente seleccionados para representar desafíos típicos que los profesionales pueden enfrentar en la industria y la investigación.

Los proyectos incluidos al final de cada capítulo permiten a los lectores aplicar los conocimientos adquiridos en situaciones del mundo real, garantizando que el aprendizaje sea relevante y aplicable. Esta estructura asegura que los estudiantes y profesionales no solo aprendan métodos numéricos, sino que también desarrollen habilidades críticas de resolución de problemas y pensamiento analítico que son indispensables en el campo de la ingeniería y las ciencias aplicadas.

TEORÍA DE ERRORES

INTEGRANTES:

CÉSPEDES MALLQUI, MIGUEL ANGEL

FLORES DIONICIO, LUZ CLARITA

DEFINICIONES

1. MÉTODOS NUMÉRICOS Y LA TEORÍA DE ERRORES

Los métodos numéricos deben ser lo suficientemente **exactos o sin sesgo** para satisfacer los requisitos de un problema particular de ingeniería. También deben ser lo suficientemente preciso para ser adecuados al diseño de la ingeniería.

En estos términos se usará el término error para representar tanto **la inexactitud como la imprecisión en los cálculos numéricos** en las imprecisiones.

FUENTE: ELABORACIÓN PROPIA- MIND MANAGER

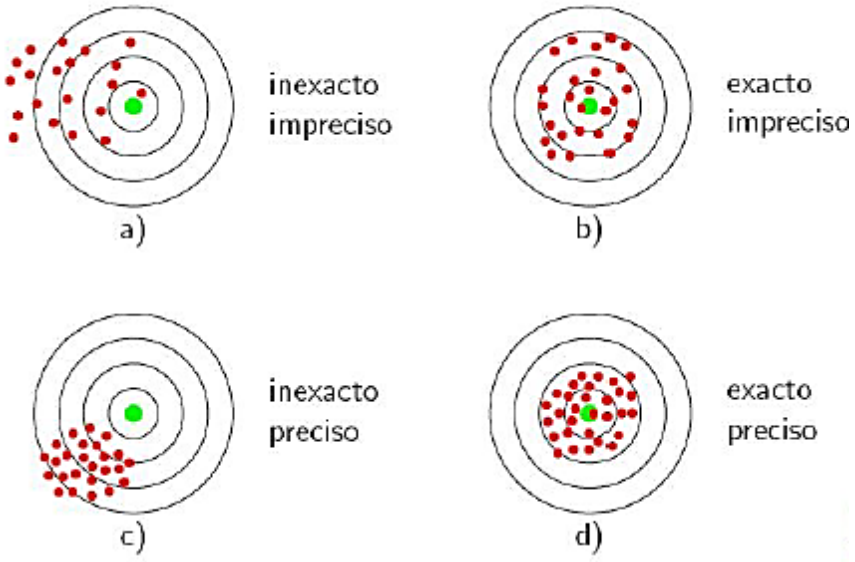
1.1 APROXIMACIÓN NUMÉRICA

- **Exactitud:** grado de concordancia entre el valor "verdadero" y el experimental. Un aparato es exacto si las medidas realizadas con él son todas muy próximas al valor "verdadero" de la magnitud medida.

- **Precisión:** concordancia entre las medidas de una misma magnitud realizadas en condiciones sensiblemente iguales. Un aparato es preciso cuando la diferencia entre diferentes mediciones de una misma magnitud es muy pequeña.

La exactitud implica, normalmente, precisión, pero la afirmación inversa no es cierta, ya que pueden existir aparatos muy precisos que posean poca exactitud, debido a errores sistemáticos, como el **"error de cero"**, etc. En general, se puede decir que es más fácil conocer la precisión de un aparato que su exactitud (básicamente, debido a la introducción del término "verdadero").

- **Sensibilidad:** valor mínimo de la magnitud que es capaz de medir. Que la sensibilidad de una balanza es de 5 mg significa que, para masas inferiores a ésta, la balanza no acusa ninguna desviación. Normalmente, se admite que la sensibilidad de un aparato viene indicada por el valor de la división más pequeña de la escala de medida. En ocasiones, de un modo erróneo, se toman como idénticos los conceptos de precisión y sensibilidad.



1.2 ERROR ABSOLUTO

Se da cuando se aproxima el valor real con un valor aproximado

$\varepsilon = \tilde{a} - a$ Donde $\tilde{a} = Valor\ aproximado$

$a = Valor\ exacto$

1.3 ERROR RELATIVO PORCENTUAL

Suele ser un mejor indicador de la precisión, es más independiente de la escala usada, y esto es una propiedad más que deseable.

$$Er = \frac{\varepsilon}{a} = \frac{\tilde{a} - a}{a} = \frac{Error}{ValorVerdadero} * 100$$

El valor de la magnitud debe de tener sólo las cifras necesarias para que su última cifra significativa sea del mismo orden decimal que la última del error absoluto, llamada cifra de acotamiento.

VALORES INCORRECTOS	VALORES CORRECTOS
3,418 ± 0,123	3,4 ± 0,1
6,3 ± 0,09	6,30 ± 0,09
46288 ± 1551	(4,6 ± 2) × 10^3
428,351 ± 0,27	428,4 ± 0,3
0,01683 ± 0,0058	0,017 ± 0,006

1.4 ERRORES EN LAS MEDIDAS FÍSICAS

Clasificación:

- Errores sistemáticos (defectos intrínsecos).
- Errores accidentales (causas fortuitas, tratamiento estadístico).

Las distintas medidas de una magnitud afectadas sólo por errores accidentales se distribuyen en torno al **"valor verdadero"** de una forma estadísticamente predecible. Cuando los errores en las medidas son accidentales, la mejor aproximación al valor verdadero es la media aritmética de los valores obtenidos.

1.5 ERROR POR REDONDEO Y TRUNCAMIENTO

Prescinde de cierto número de cifras significativas y realiza un ajuste sobre la última cifra.

NO. DE DÍGITOS	CORTE	REDONDEO
Dos	3.1	3.1
Tres	3.14	3.14
Cuatro	3.141	3.142
Cinco	3.1415	3.1416
Seis	3.14159	3.14159
Siete	3.141592	3.141593
Ocho	3.1415926	3.1415927

1.6 MÉTODOS ITERATIVOS

SERIE TAYLOR:

Aproximación de funciones mediante una serie de potencias o suma de potencias enteras de polinomios como llamados términos de la serie, dicha suma se calcula a partir de las derivadas de la función para un determinado valor.

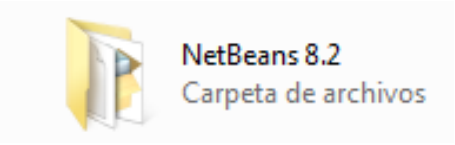
SERIE MACLAURIN:

Permite determinar de forma exacta el número de Euler (base de los logaritmos naturales

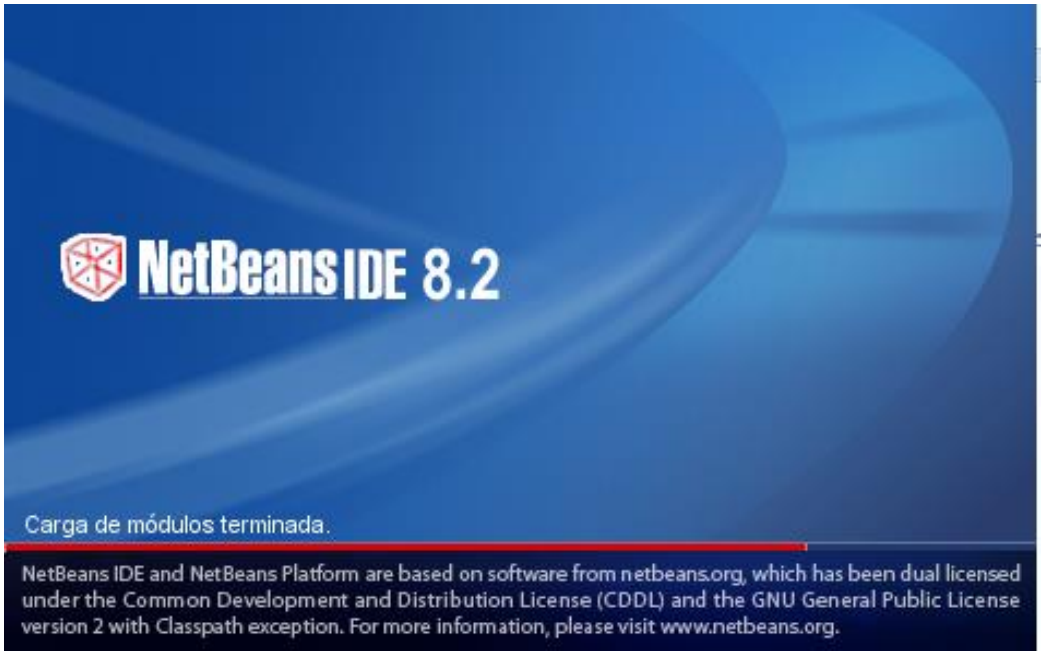
2. SOFTWARE TEORÍA DE ERRORES

2.1. CONTENIDO

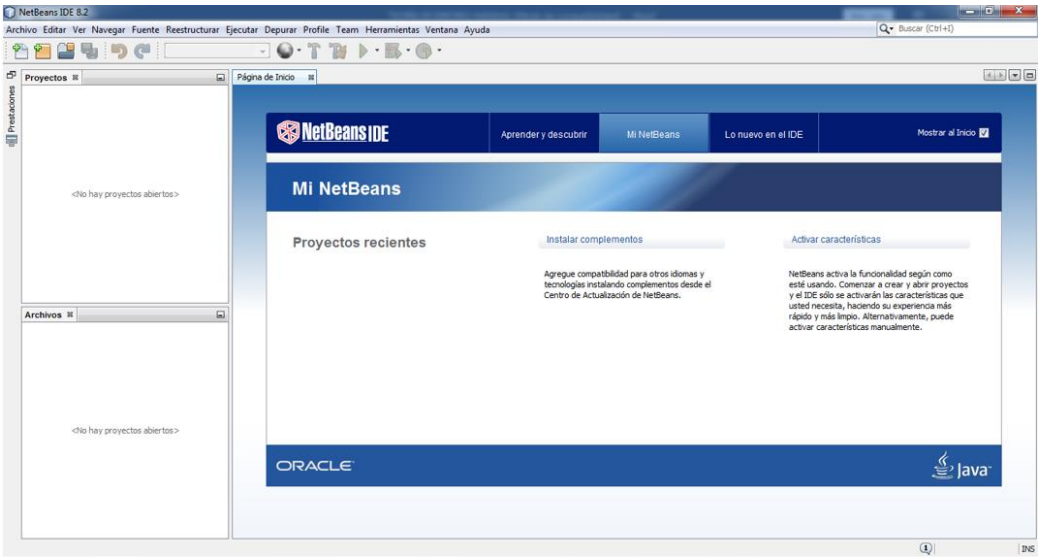
El software está hecho en el programa Java Netbeans 8.2. Para ello iniciamos instalando dicho programa.



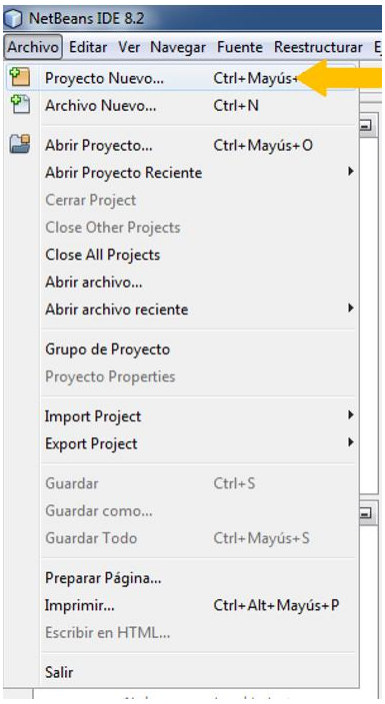
Una vez instalado el programa ejecutamos el programa.



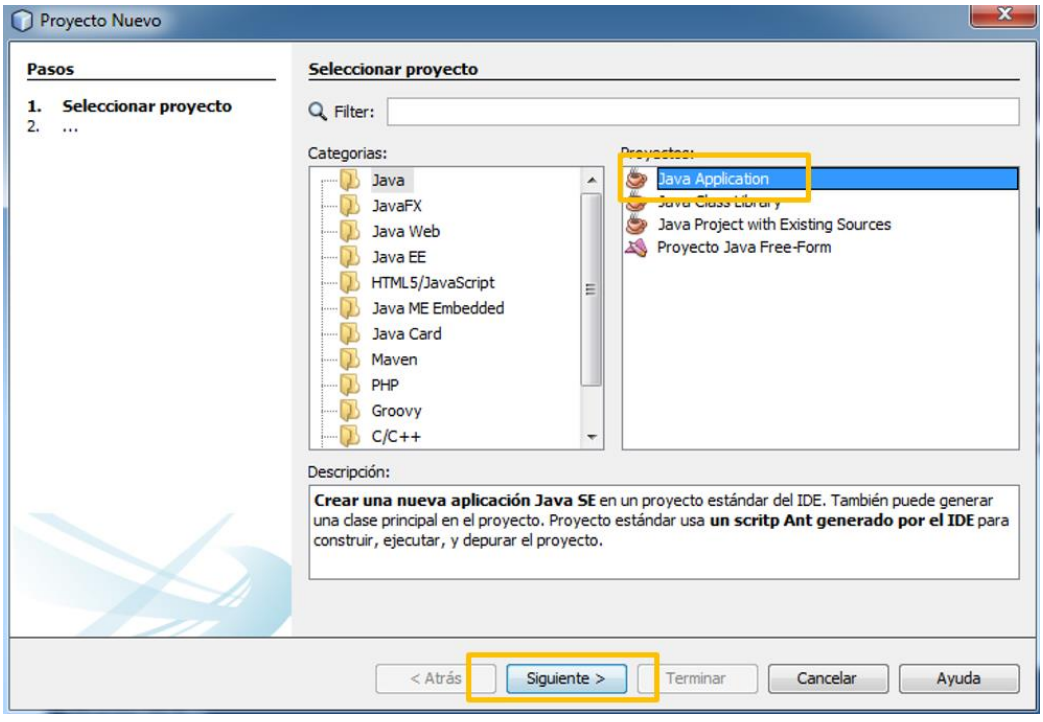
Observaremos la siguiente ventana, lo cual indica que el programa ya inició.



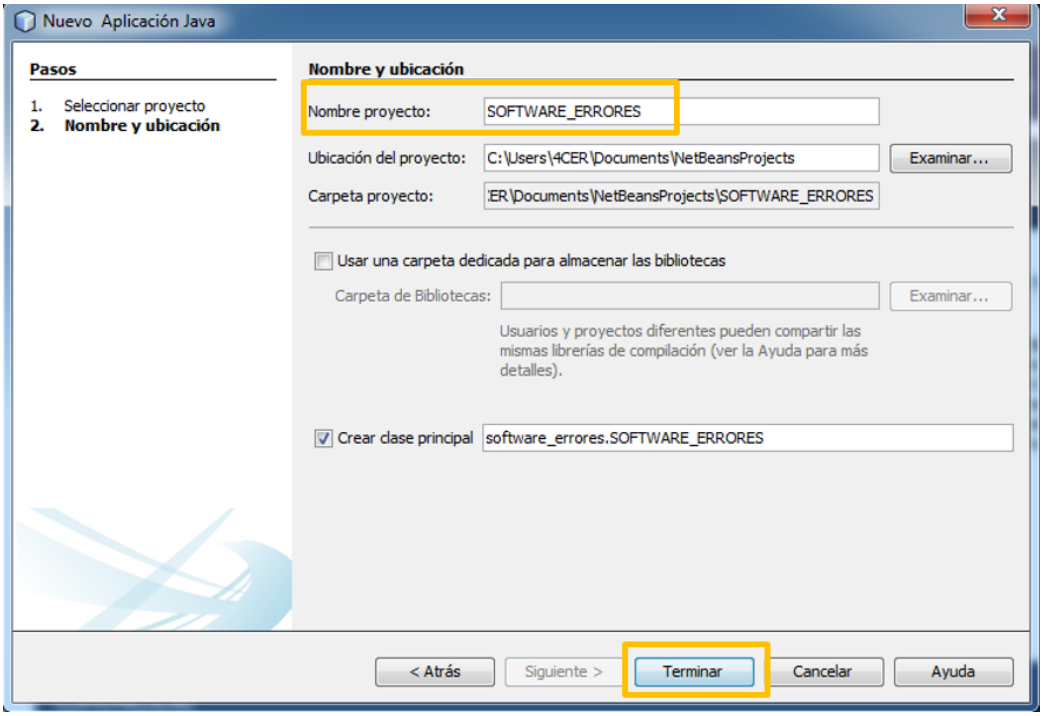
Una vez iniciado el programa, para crear nuestro software damos clic en la opción ***ARCHIVO > PROYECTO NUEVO***.



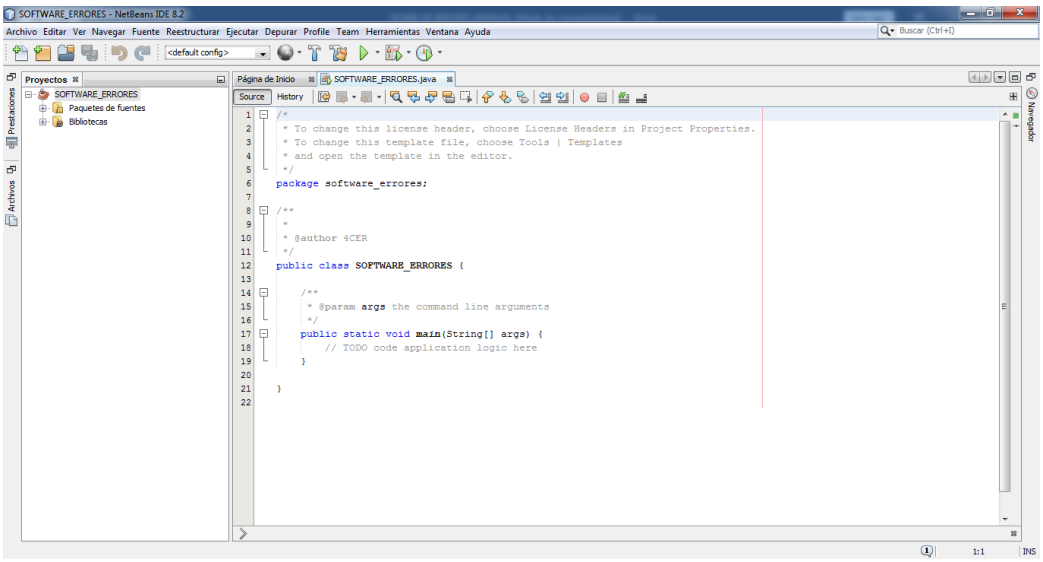
Observaremos la siguiente ventana, elegiremos la opción ***JAVA APLICATION > SIGUIENTE.***

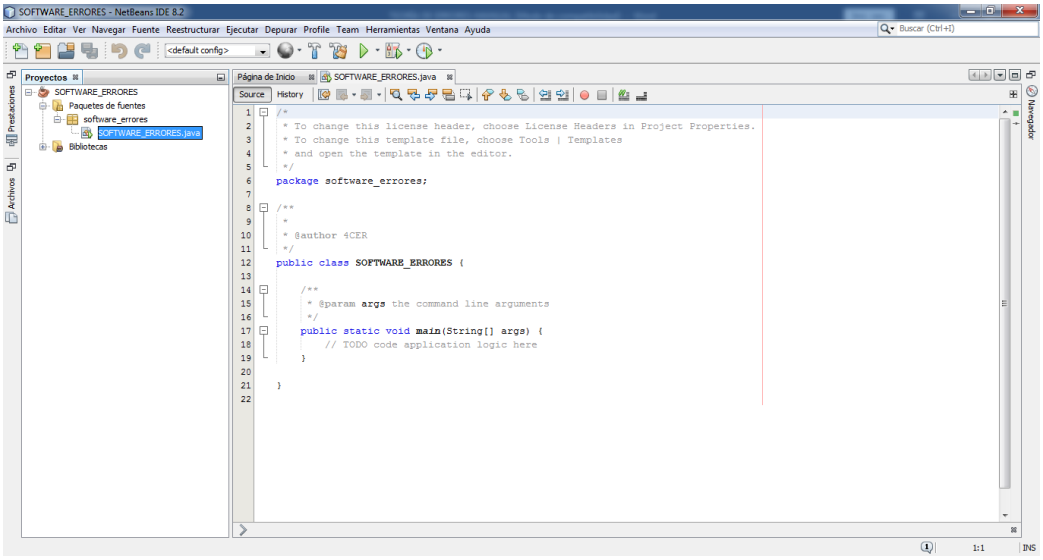


Luego pondremos el nombre de nuestro proyecto ***"SOFTWARE_ERRORES" > TERMINAR.***

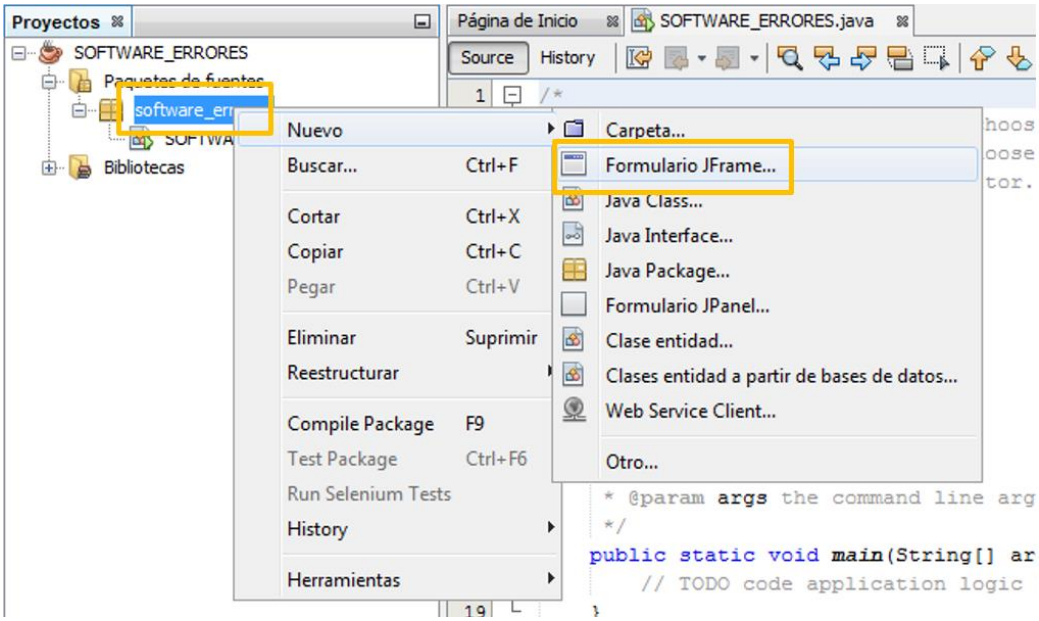


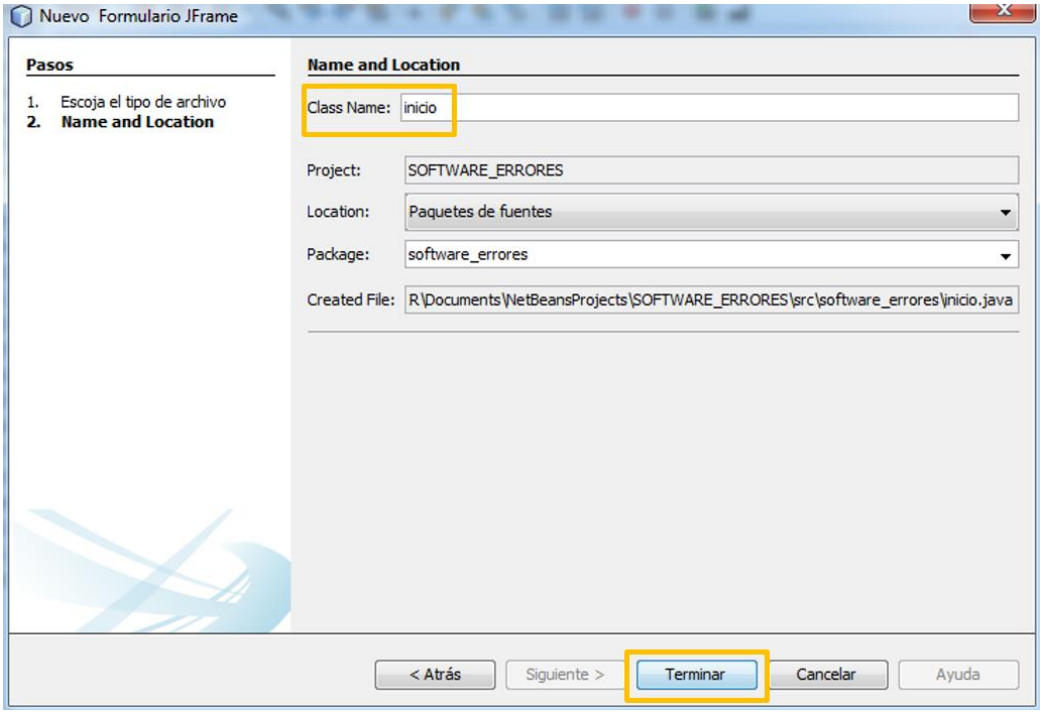
Observaremos la siguiente ventana.

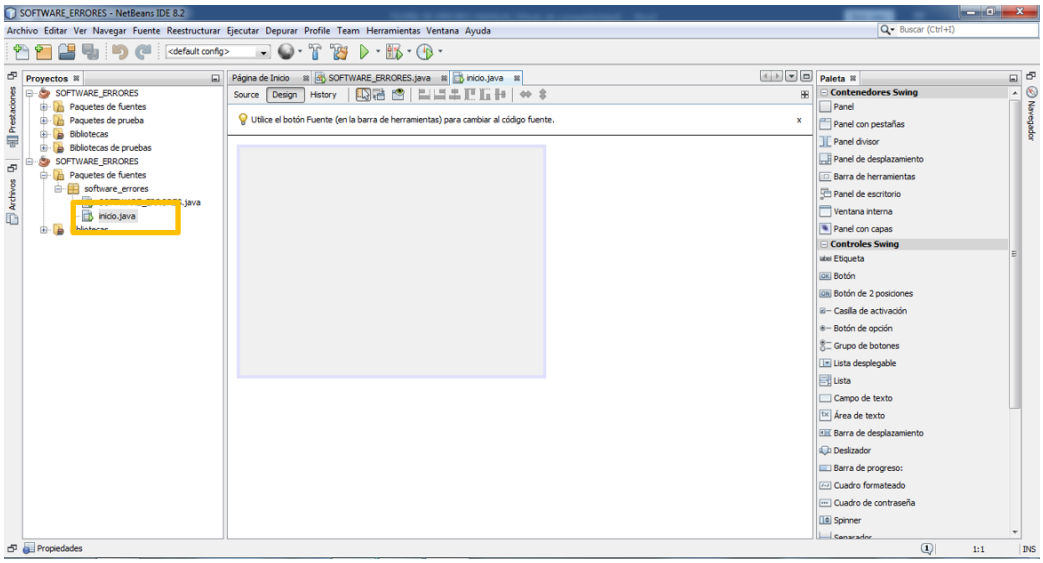




Para crear nuestra ventana de inicio a la que llamaremos, creamos un JFrame a la que llamaremos INICIO donde se ubicará las dos opciones de nuestra ventana. Para ello damos clic en ***SOFTWARE_ERRORES>NUEVO>FORMULARIO JFRAME***.

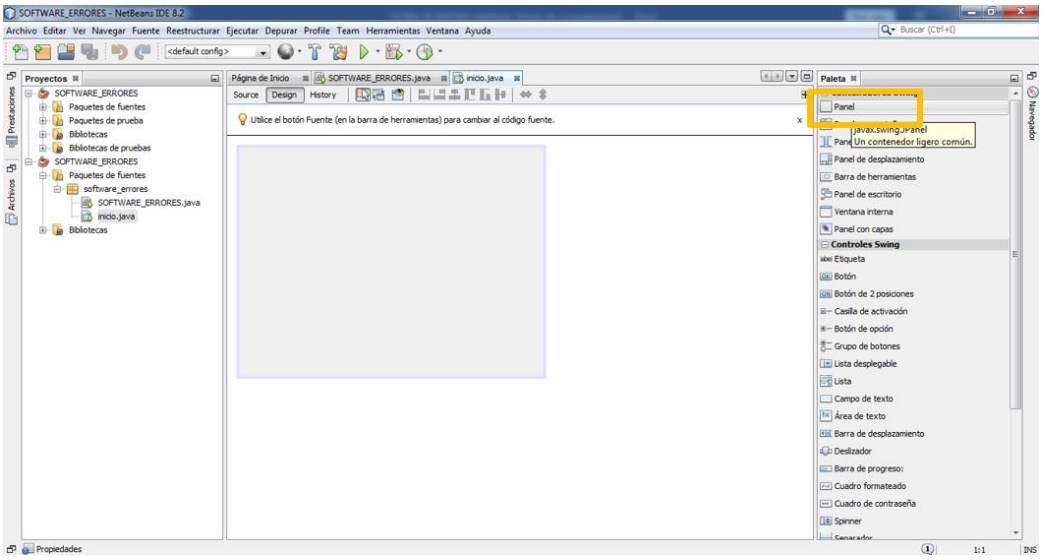


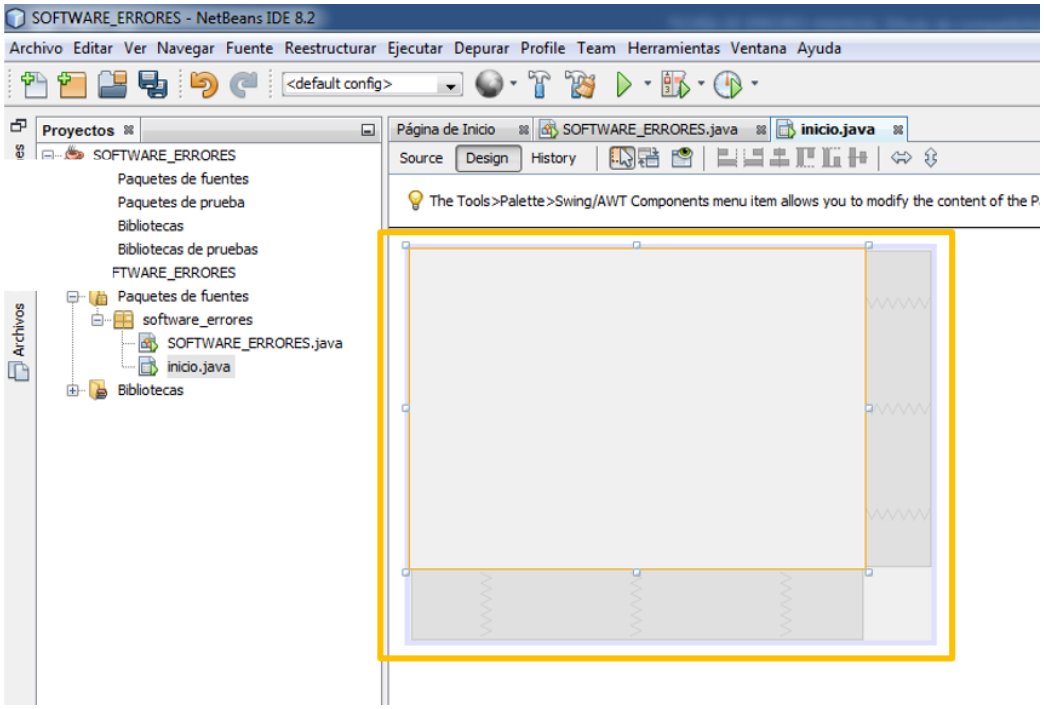




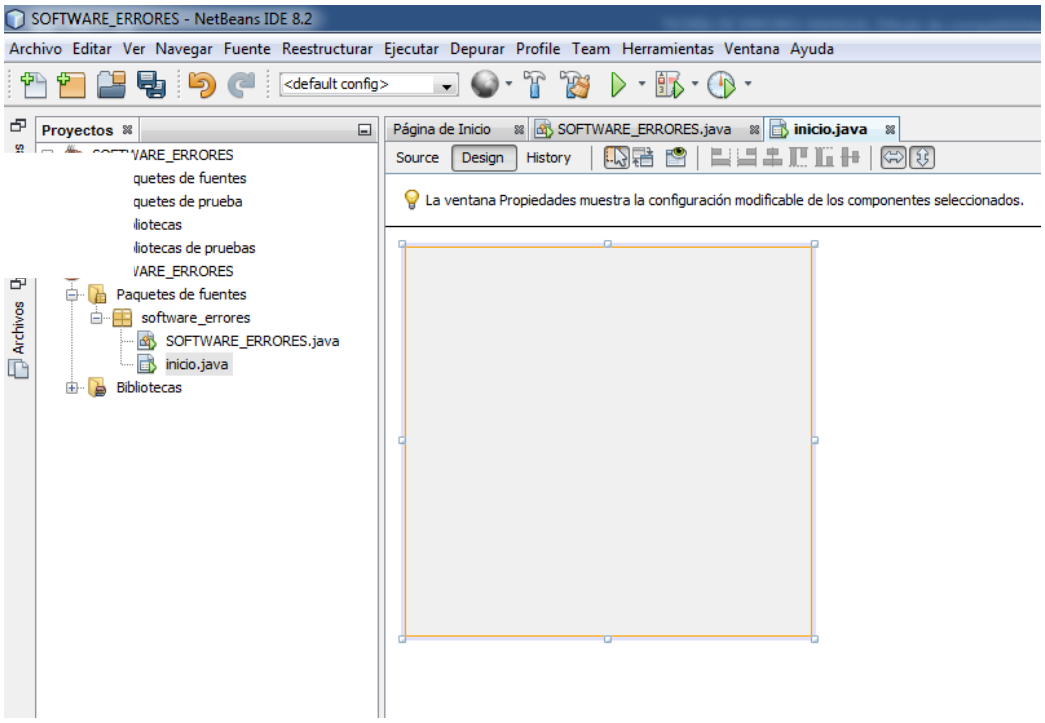
Para construir nuestra ventana de inicio tal como lo observamos a continuación, realizamos las siguientes operaciones. Agregamos de la PALETA, los siguientes componentes:

En primer lugar, insertamos un Panel y arrastramos hacia el JFrame creado para luego posteriormente poner un fondo de color a nuestra ventana de inicio.

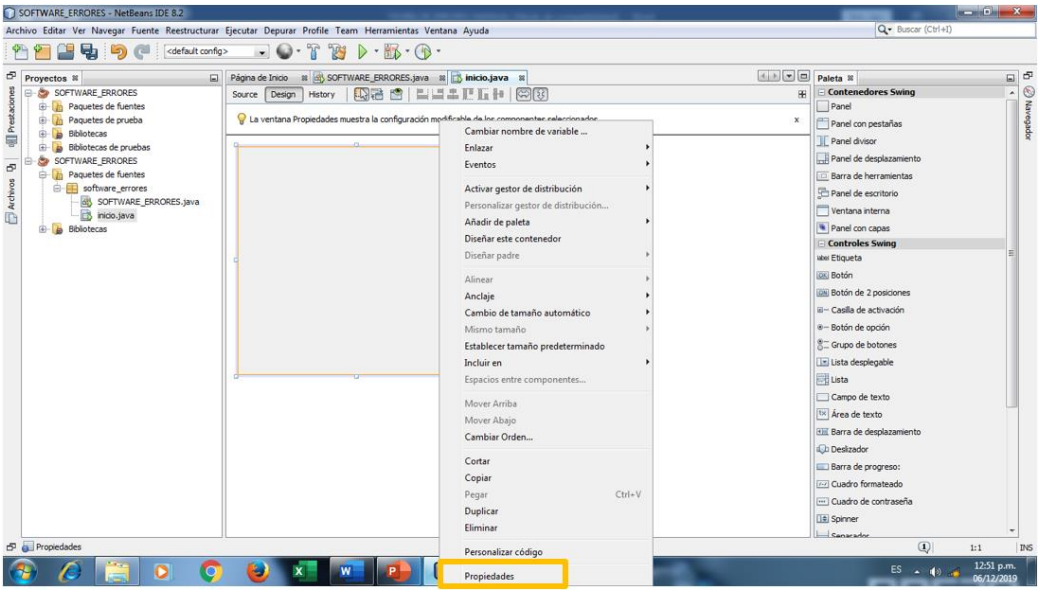




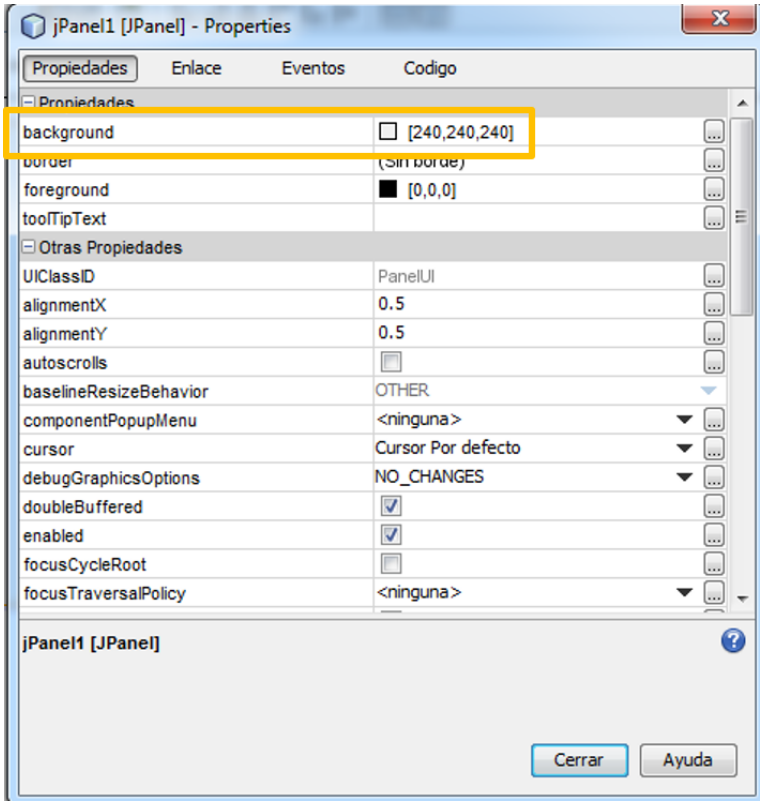
Y ajustamos a nuestra ventana de ***INICIO***.

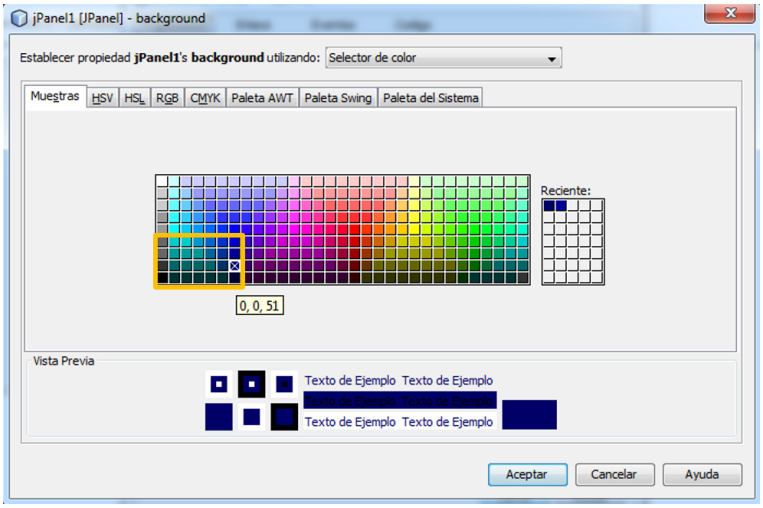


Y para darle el color de fondo de nuestra ventana inicial, realizaremos las siguientes operaciones, para ello damos clic derecho en el Panel insertado y obtendremos las siguientes opciones, donde haremos clic en la opción propiedades para poder cambiar el color del fondo de nuestro panel

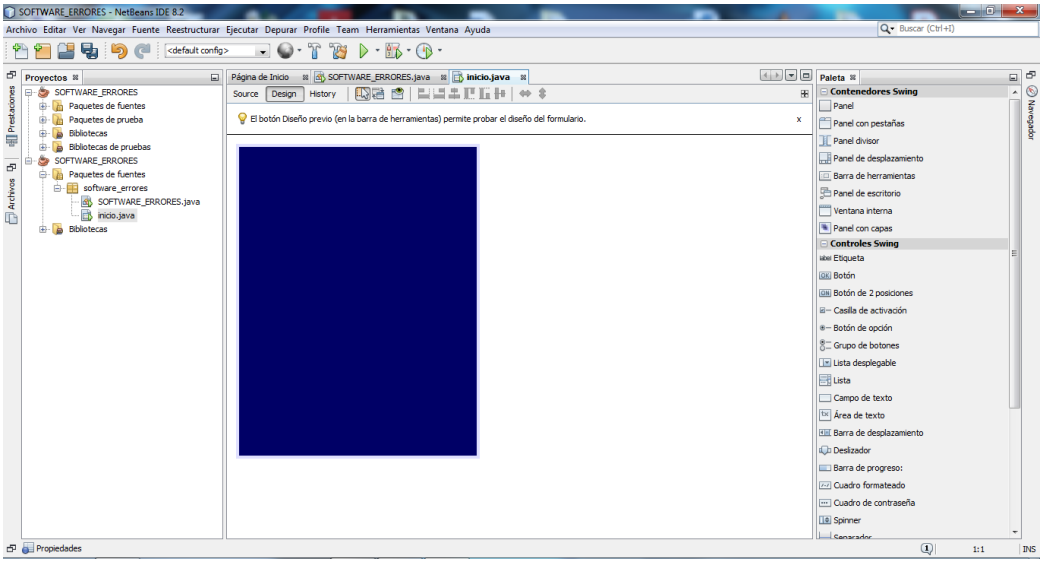


Luego observaremos lo siguiente y daremos clic en la opción ***BACKGROUND*** para cambiar el color del fondo, y posteriormente elegimos el color, y por último damos clic en la opción ***ACEPTAR***.

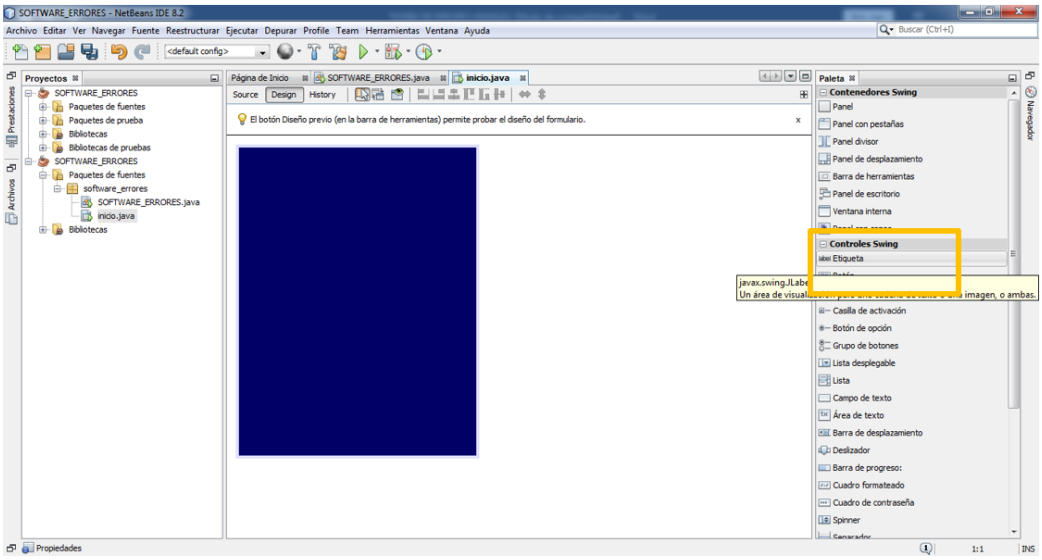


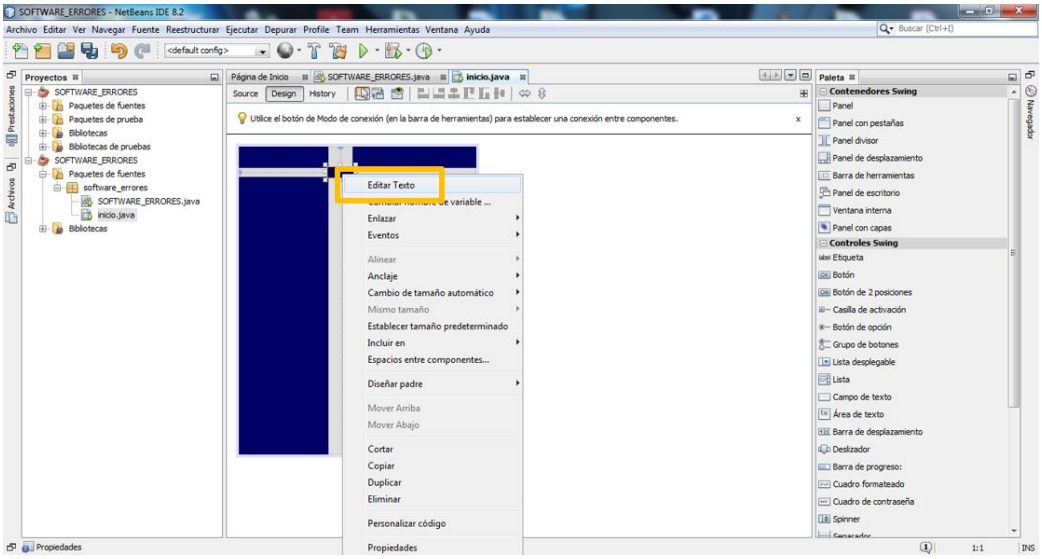


Y tendremos nuestro Panel del color azul tal y como lo elegimos.

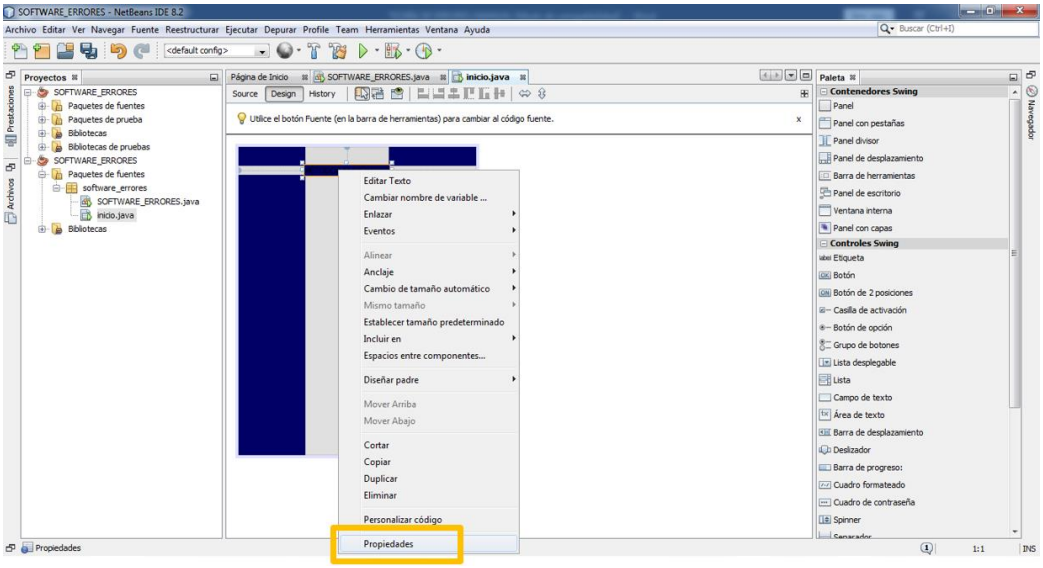


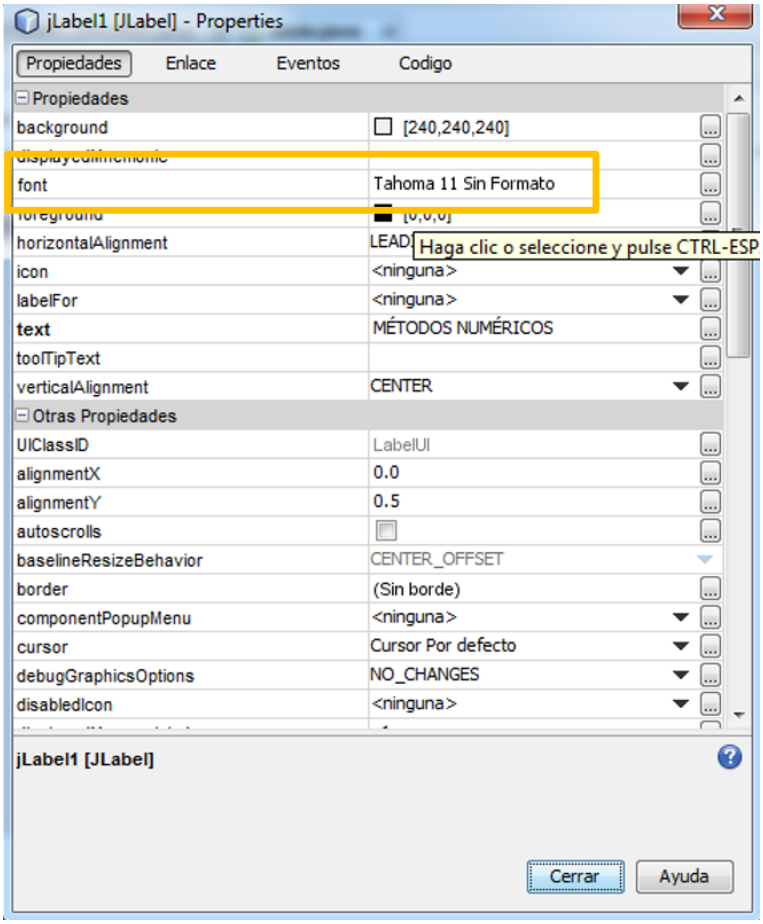
Para insertar el titulo MÉTODOS NUMÉRICOS, debemos de arrastrar de la paleta la herramienta ETIQUETA (JLABEL), luego editamos el texto y colocamos el titulo ***MÉTODOS NUMÉRICOS***.

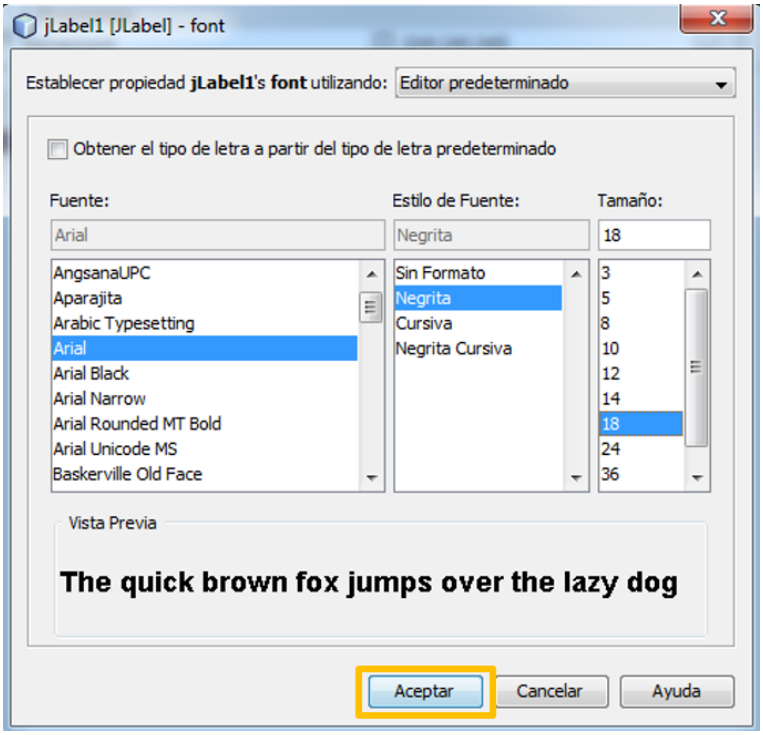




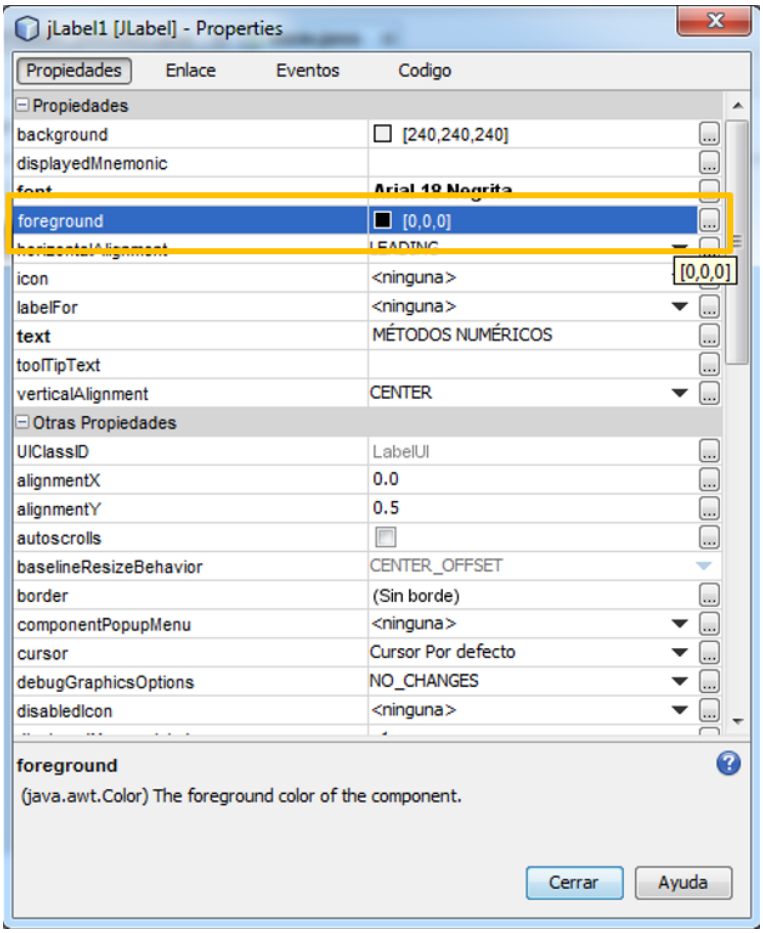
Para cambiar el formato de las letras del título MÉTODOS NUMÉRICOS, realizamos la siguiente operación. Clic derecho en ***JLABEL>PROPIEDADES> FONT***.

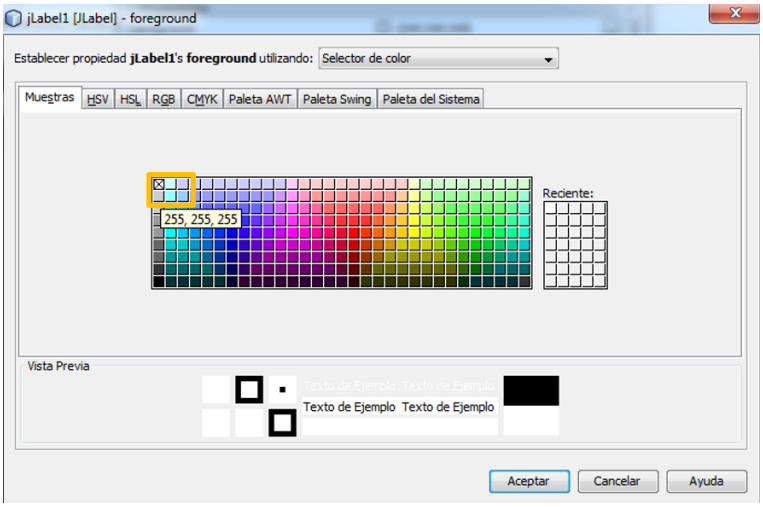

SOFTWARE_ERRORES - NetBeans IDE 8.2
Archivo Editar Ver Navegar Fuente Reestructurar Ejecutar Depurar Profile Team Herramientas Ventana Ayuda
Proyectos
SOFTWARE_ERRORES
Paquetes de fuentes
Paquetes de prueba
Bibliotecas
Bibliotecas de pruebas
SOFTWARE_ERRORES
Paquetes de fuentes
software_errores
SOFTWARE_ERRORES.java
inicio.java
Bibliotecas
Página de Inicio
SOFTWARE_ERRORES.java
inicio.java
Source
Design
History
Utilice el botón Fuente (en la barra de herramientas) para cambiar al código fuente.
Editar Texto
Cambiar nombre de variable ...
Enlazar
Eventos
Alinear
Anclaje
Cambio de tamaño automático
Mismo tamaño
Establecer tamaño predeterminado
Incluir en
Espacios entre componentes...
Diseñar padre
Mover Arriba
Mover Abajo
Cortar
Copiar
Duplicar
Eliminar
Personalizar código
Propiedades
Paleta
Contenedores Swing
Panel
Panel con pestañas
Panel divisor
Panel de desplazamiento
Barra de herramientas
Panel de escritorio
Ventana interna
Panel con capas
Controles Swing
Etiqueta
Botón
Botón de 2 posiciones
Casilla de activación
Botón de opción
Grupo de botones
Lista desplegable
Lista
Campo de texto
Área de texto
Barra de desplazamiento
Deslizador
Barra de progreso:
Cuadro formateado
Cuadro de contraseña
Spinner
Propiedades
1:1
INS


jLabel1 [JLabel] - Properties
Propiedades
Enlace
Eventos
Codigo
Propiedades
background
[240,240,240]
font
Tahoma 11 Sin Formato
horizontalAlignment
LEAD
Haga clic o seleccione y pulse CTRL-ESP
icon
<ninguna>
labelFor
<ninguna>
text
MÉTODOS NUMÉRICOS
toolTipText
verticalAlignment
CENTER
Otras Propiedades
UIClassID
LabelUI
alignmentX
0.0
alignmentY
0.5
autoscrolls
baselineResizeBehavior
CENTER_OFFSET
border
(Sin borde)
componentPopupMenu
<ninguna>
cursor
Cursor Por defecto
debugGraphicsOptions
NO_CHANGES
disabledIcon
<ninguna>
jLabel1 [JLabel]
Cerrar
Ayuda



Y para cambiar el fondo de nuestro Jlabel, realizamos la siguiente operación ***PROPIEDADES > FOREGROUND > OPCION DE COLOR > ACEPTAR.***

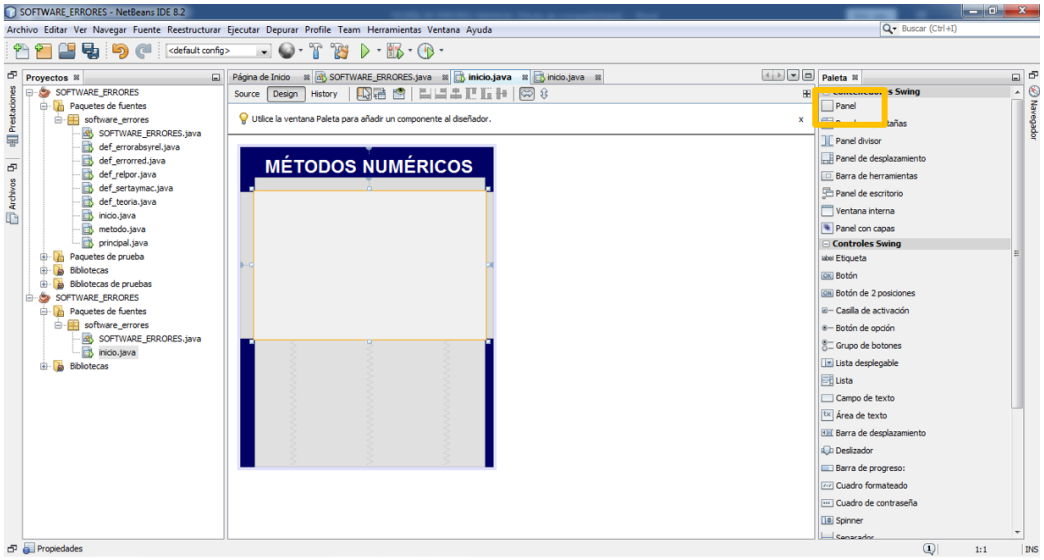




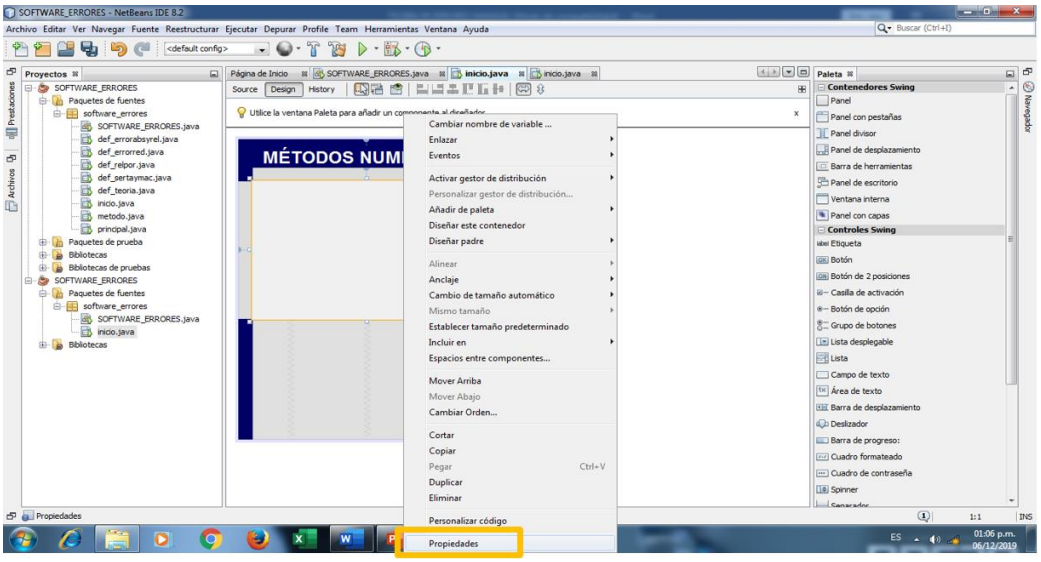
Tal como observamos a continuación.

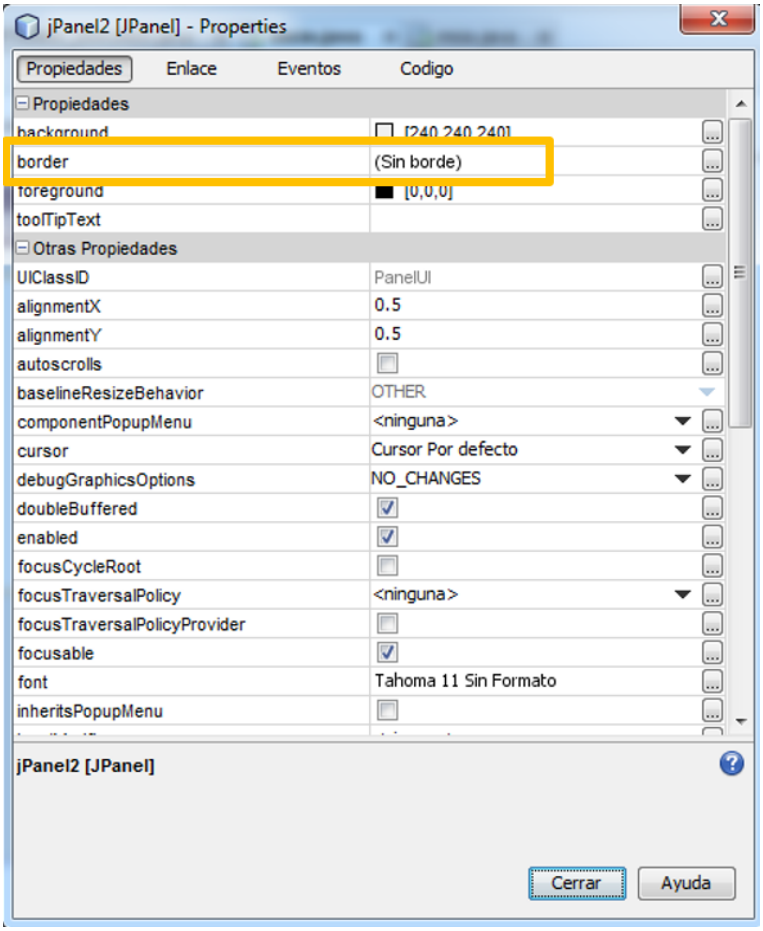
Luego para añadir las opciones y los botones de nuestra ventana de inicio, realizamos lo siguiente: para la primera opción de DEFINICIONES creamos lo siguiente:

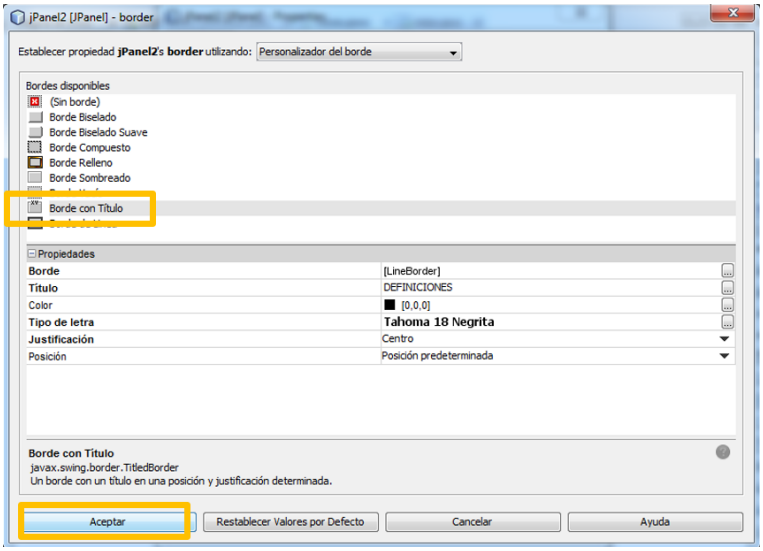
Creamos un JPanel



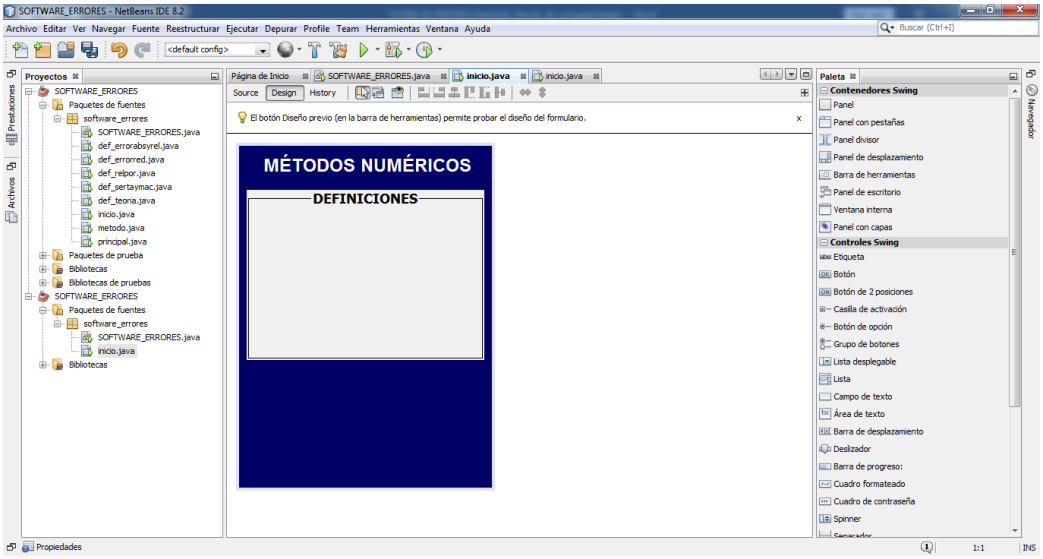
Luego damos clic en la opción ***PROPIEDADES > BORDER > BORDER CON TÍTULO> ACEPTAR***.



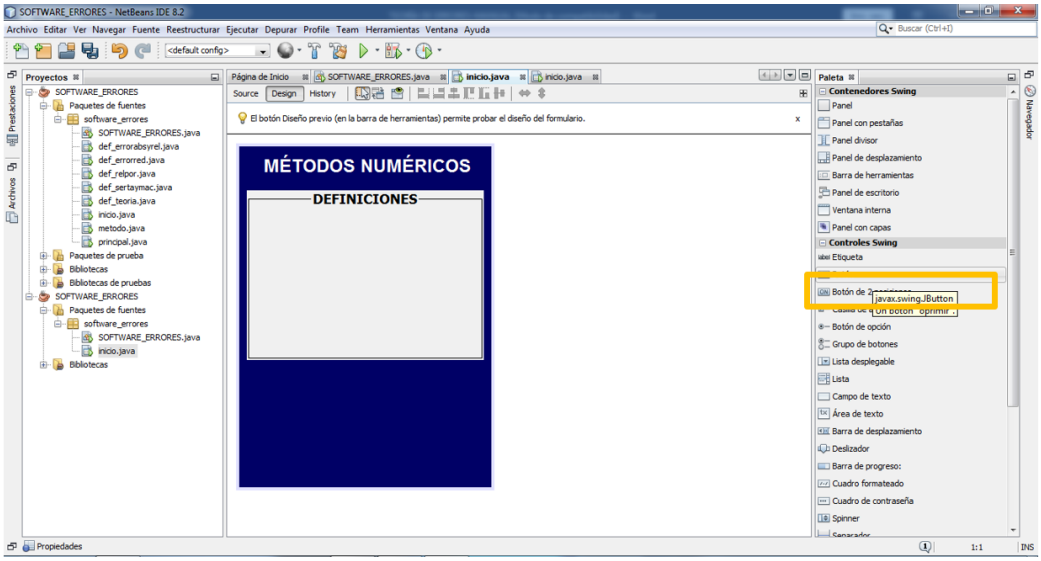


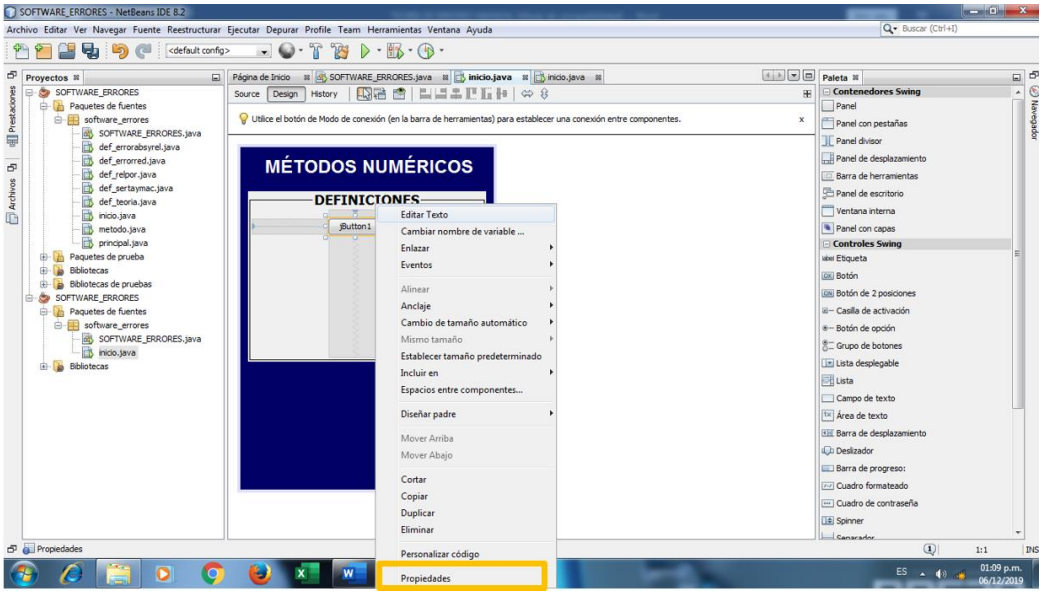


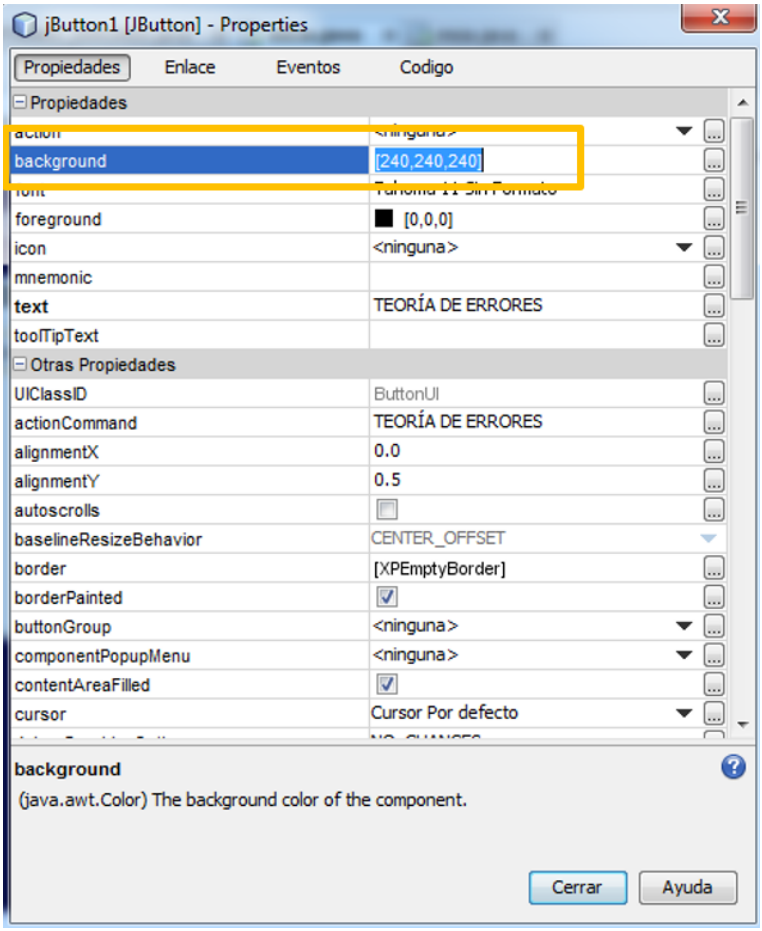
Tal como observamos en la siguiente ventana.

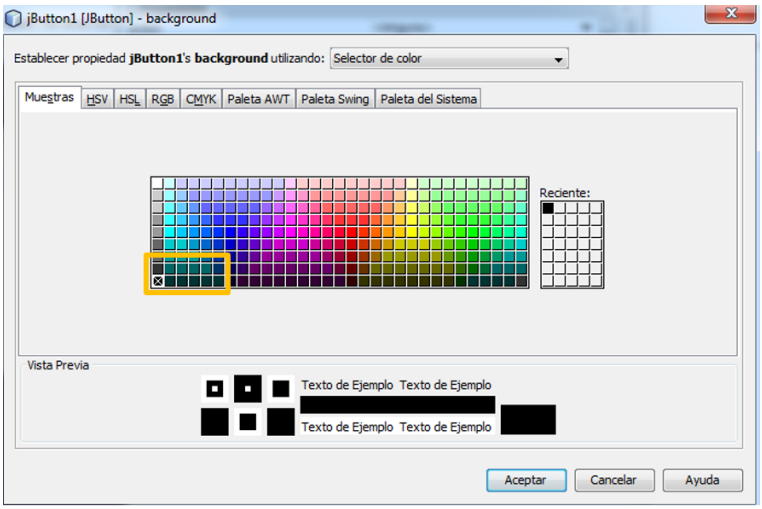


Luego de las operaciones realizadas, agregamos los botones (JBUTTON) que nos servirán como opciones, para ello insertaremos botones y colocaremos los siguientes títulos, editamos los títulos y cambiamos los fondos de colores de los botones.

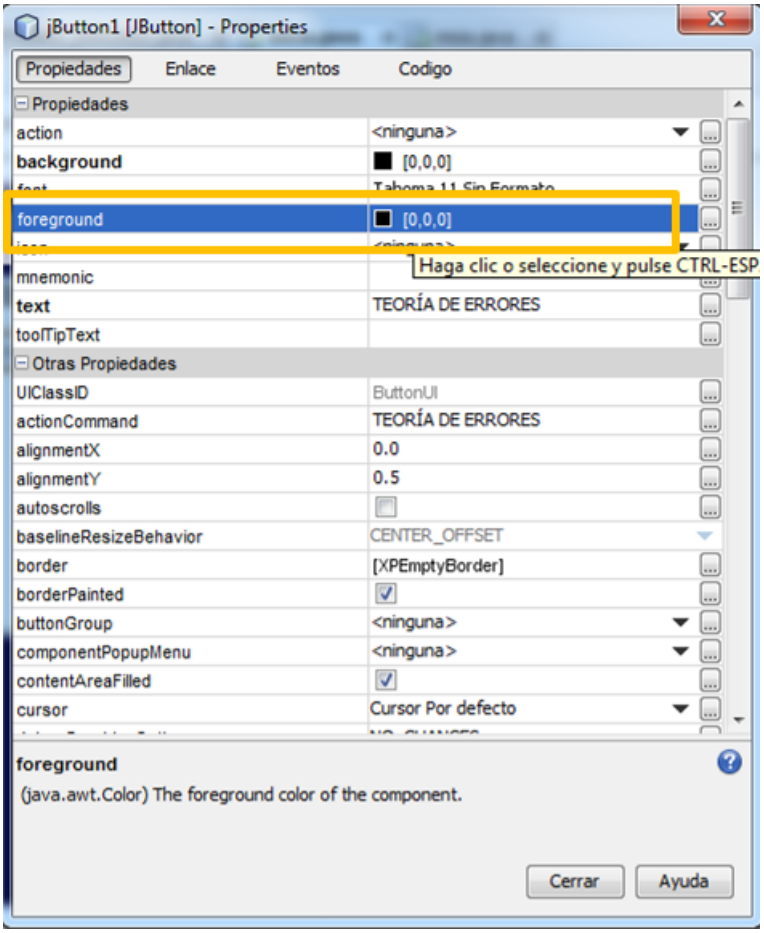

SOFTWARE_ERRORES - NetBeans IDE 8.2
MÉTODOS NUMÉRICOS
DEFINICIONES
javax.swing.JButton

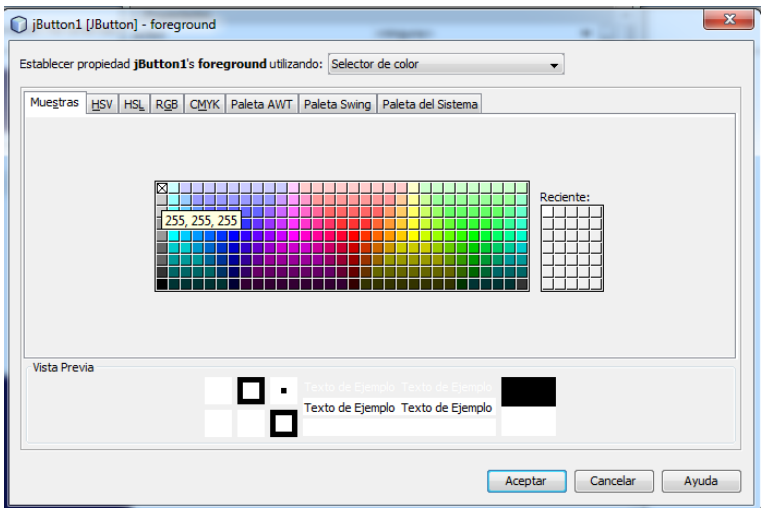

SOFTWARE_ERRORES - NetBeans IDE 8.2
MÉTODOS NUMÉRICOS
DEFINICIONES
jButton1
Editar Texto
Cambiar nombre de variable ...
Enlazar
Eventos
Alinear
Anclaje
Cambio de tamaño automático
Mismo tamaño
Establecer tamaño predeterminado
Incluir en
Espacios entre componentes...
Diseñar padre
Mover Arriba
Mover Abajo
Cortar
Copiar
Duplicar
Eliminar
Personalizar código
Propiedades



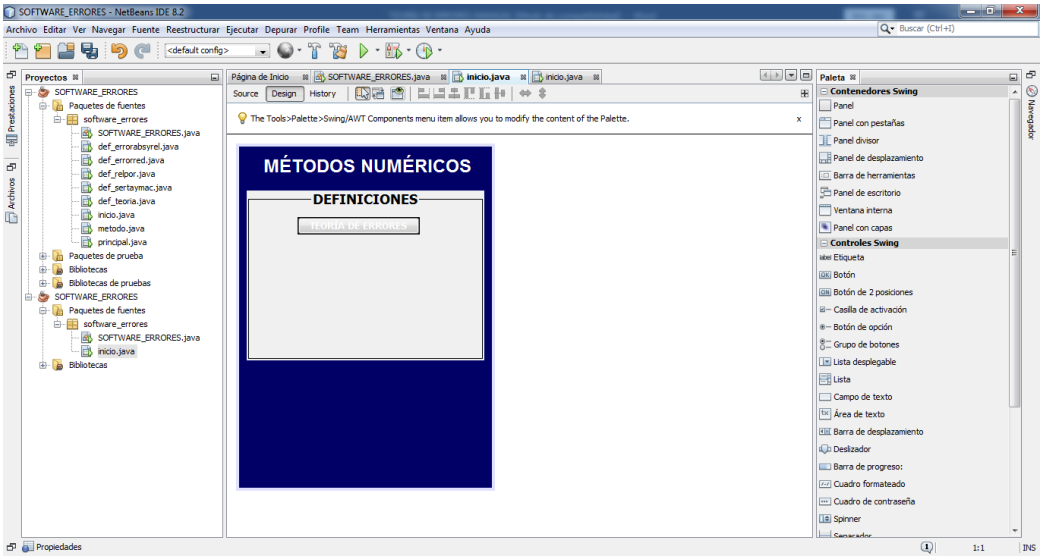


Y para cambiar el fondo de las letras, realizamos las siguientes operaciones, dando clic en la opción ***PROPIEDADES > FOREGROUND > COLOR ELEGIDO > ACEPTAR***.

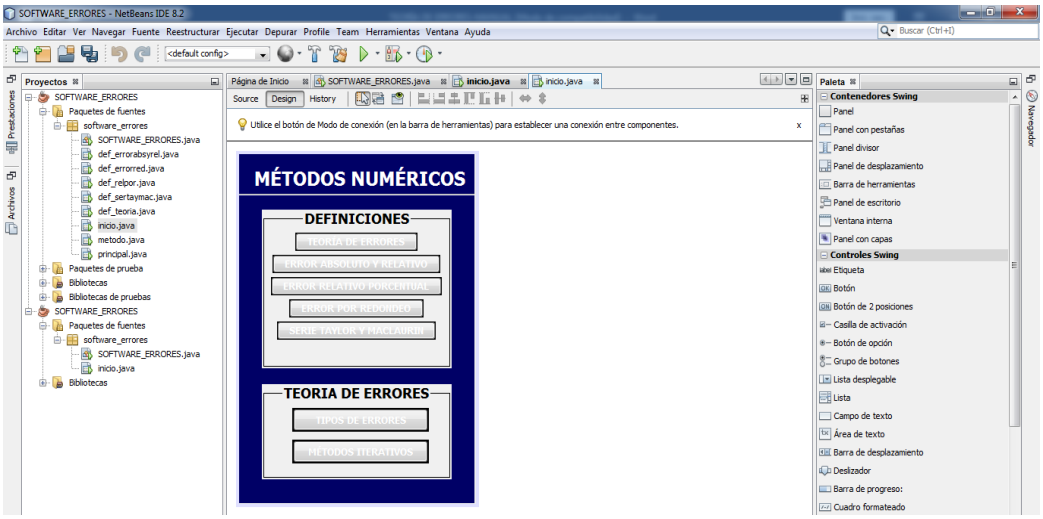




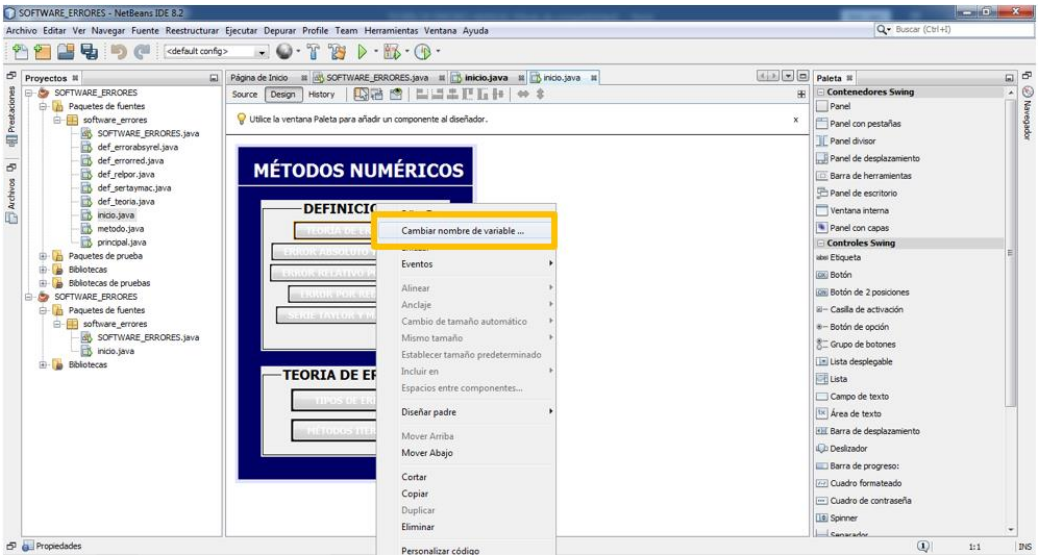
Tal como observamos en la siguiente ventana.

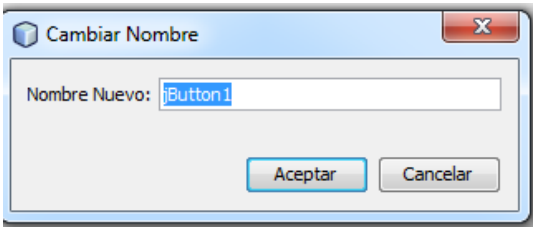


Realizamos las mismas operaciones para los botones **ERROR ABSOLUTO Y RELATIVO, ERROR RELATIVO PORCENTUAL, ERROR POR REDONDEO, SERIE DE TAYLOR Y MACLAURIN**, y también para la segunda opción de **TEORIA DE ERRORES** y los botones **TIPOS DE ERRORES Y MÉTODOS ITERATIVOS** tal como observamos a continuación:



Para ellos tendremos en cuenta el nombre de las variables de los botones para poder codificarlos posteriormente. Para ello realizamos las siguientes operaciones, clic derecho sobre el botón ***TEORÍA DE ERRORES > CAMBIAR NOMBRE DE VARIABLE.***





Los nombres de las variables para cada uno de los botones son los siguientes

TEORÍA DE ERRORES = jButton1

ERROR ABSOLUTO Y RELATIVO = jButton2

ERROR RELATIVO PORCENTUAL= jButton3

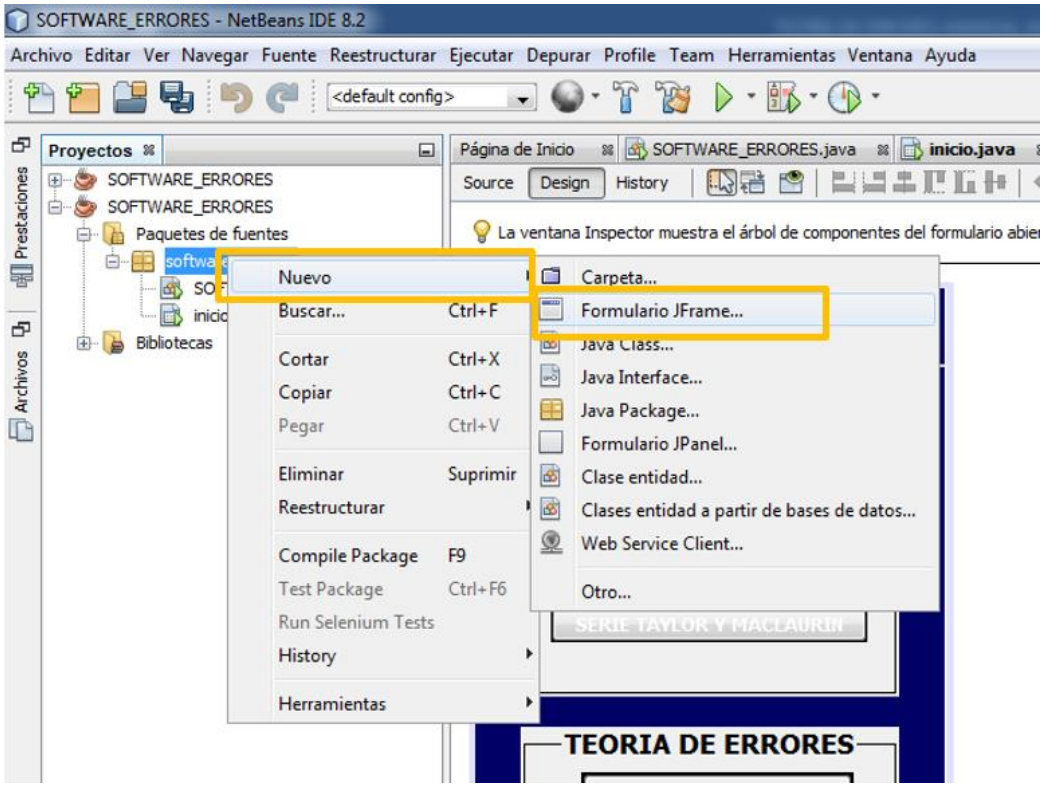
ERROR POR REDONDEO= jButton4

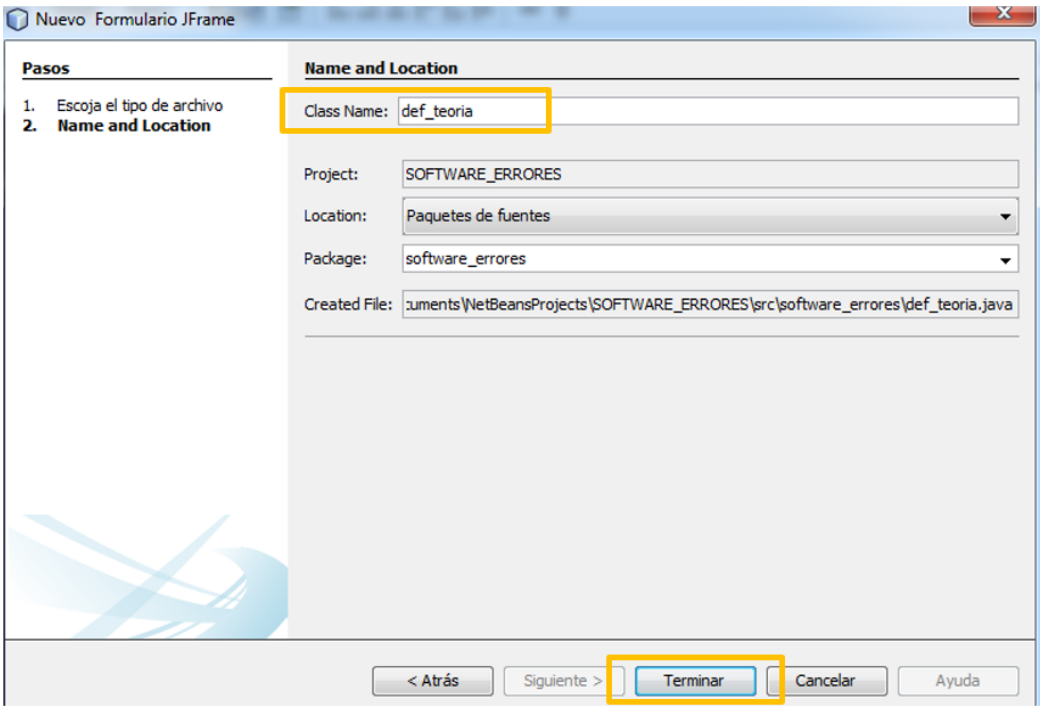
SERIE DE TAYLOR Y MACLAURIN= jButton5

TIPOS DE ERRORES = btn_inicio

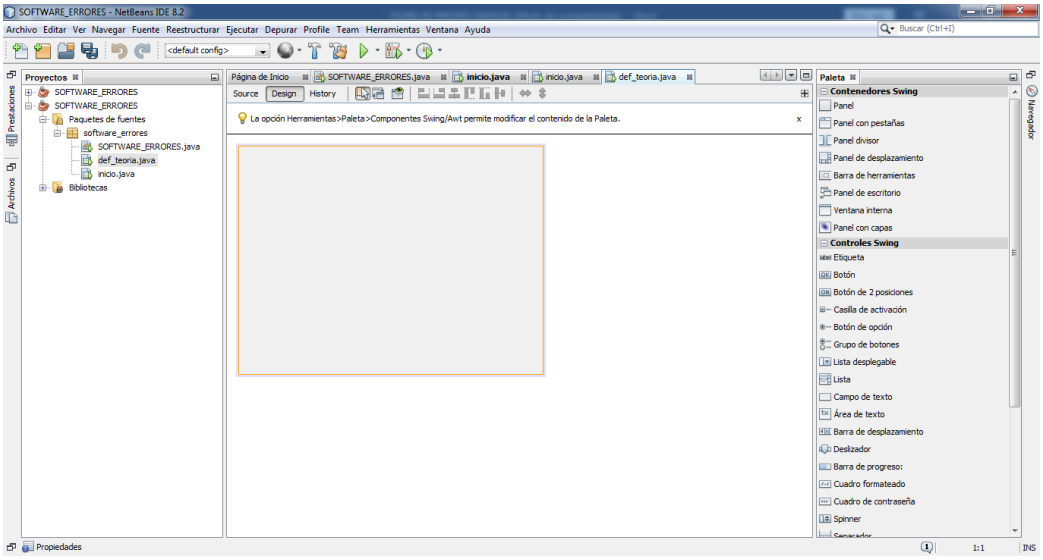
MÉTODOS ITERATIVOS = btn_inicio1

Concluido con esta operación, creamos un JFrame para cada uno de los botones.

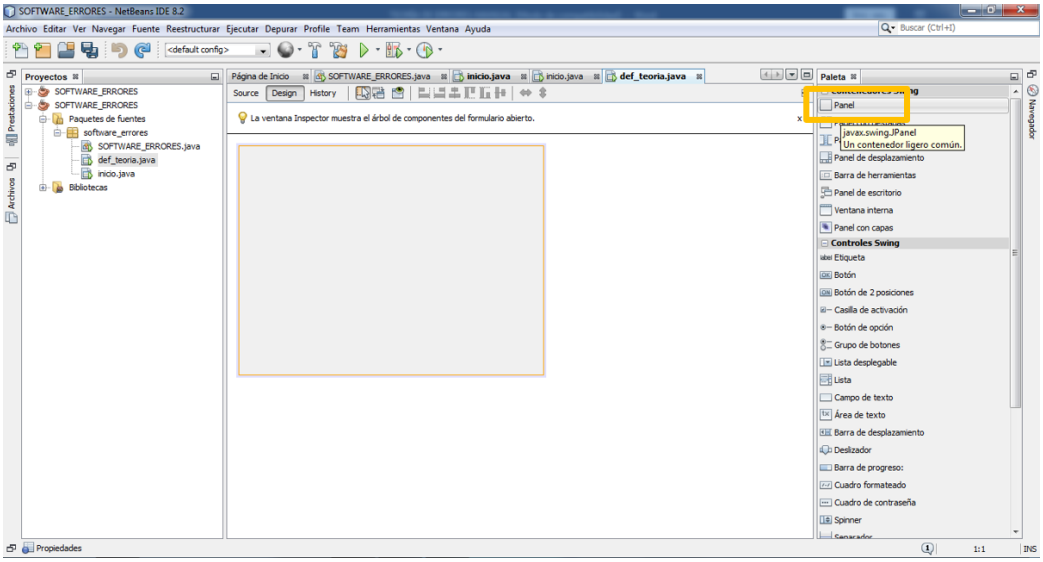


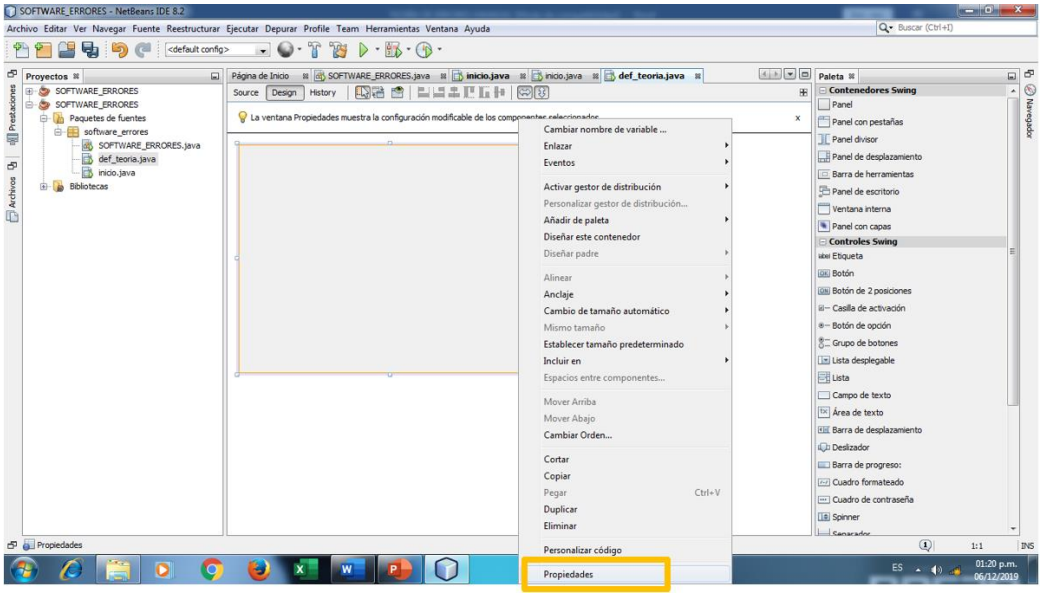


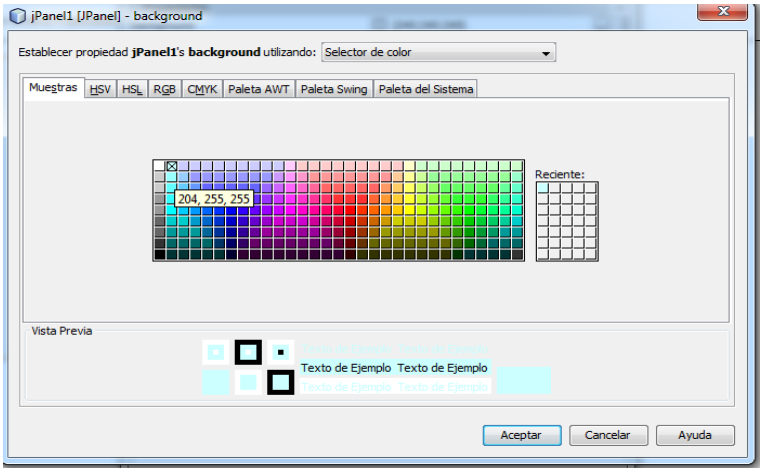
Tal como observamos a continuación:



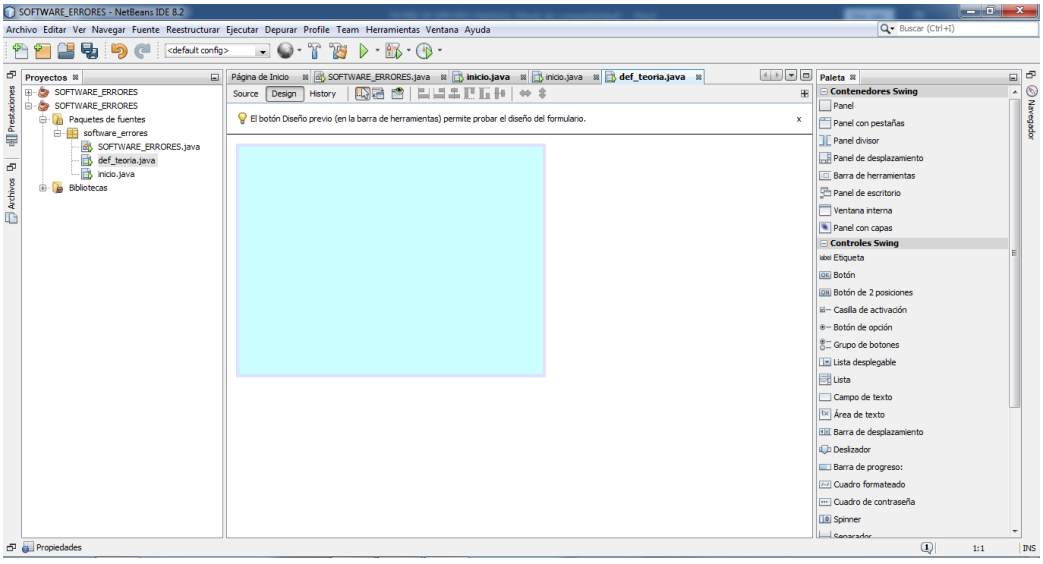
Añadimos un Jpanel que nos servirá como el fondo de nuestra ventana, donde podremos cambiar de color, para ello realizamos, lo siguiente:



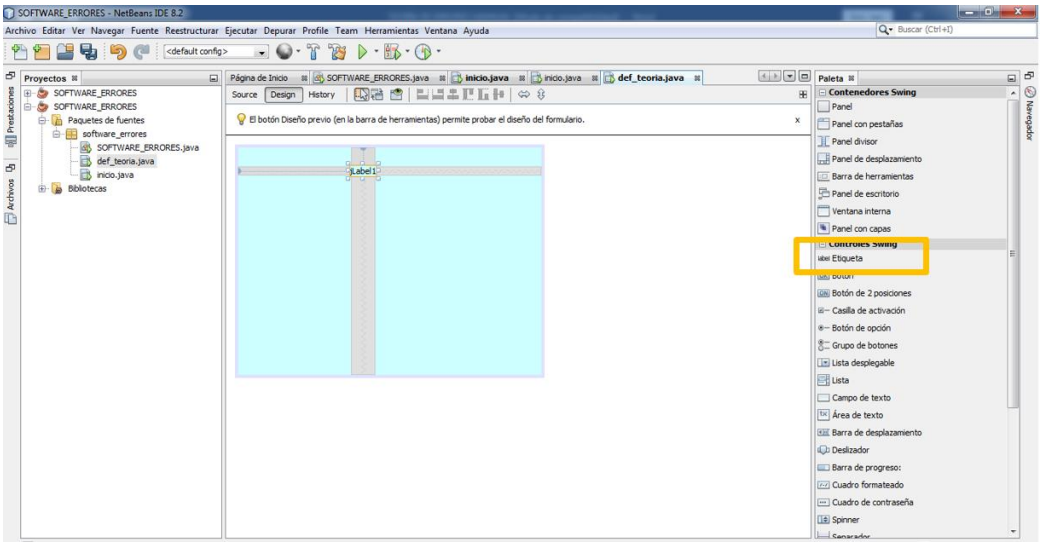


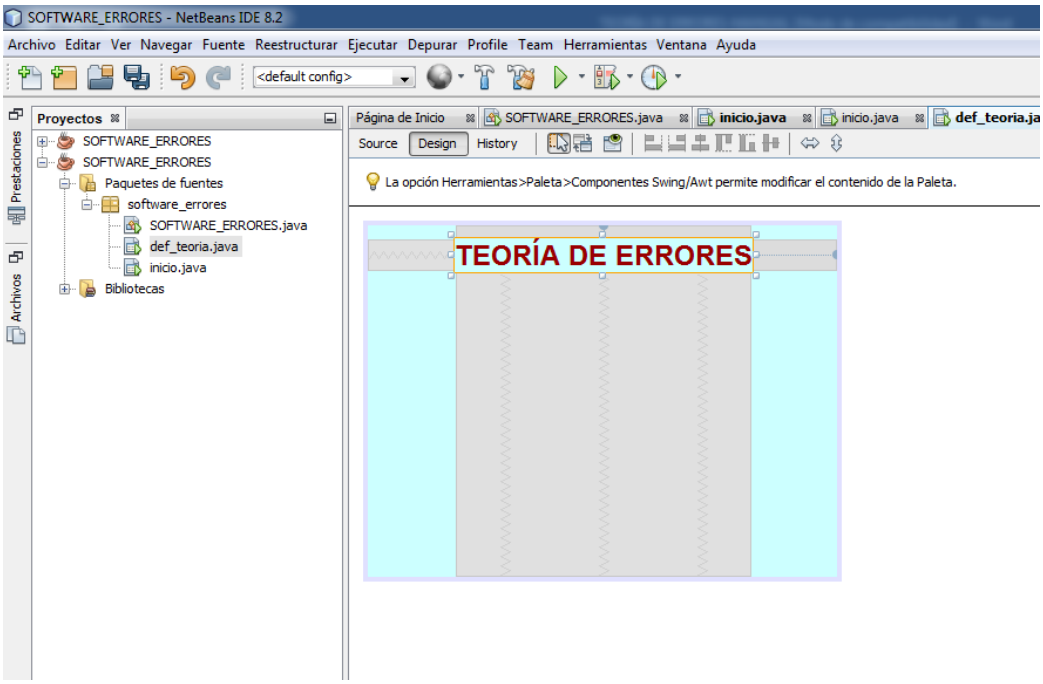


Tal como observamos en la siguiente ventana:

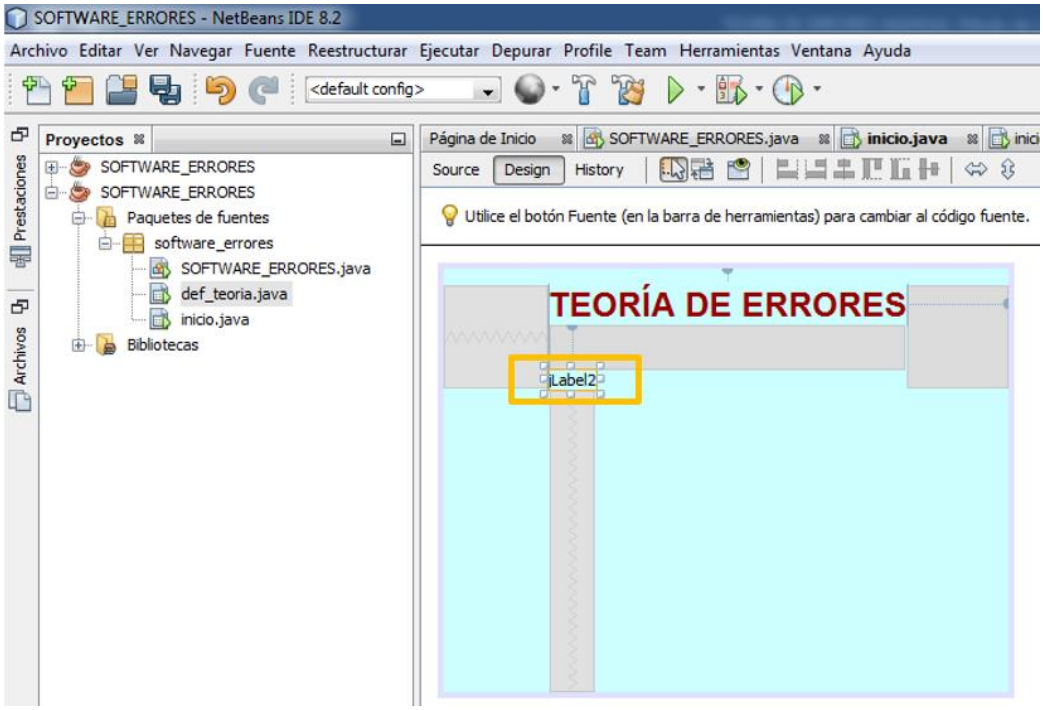


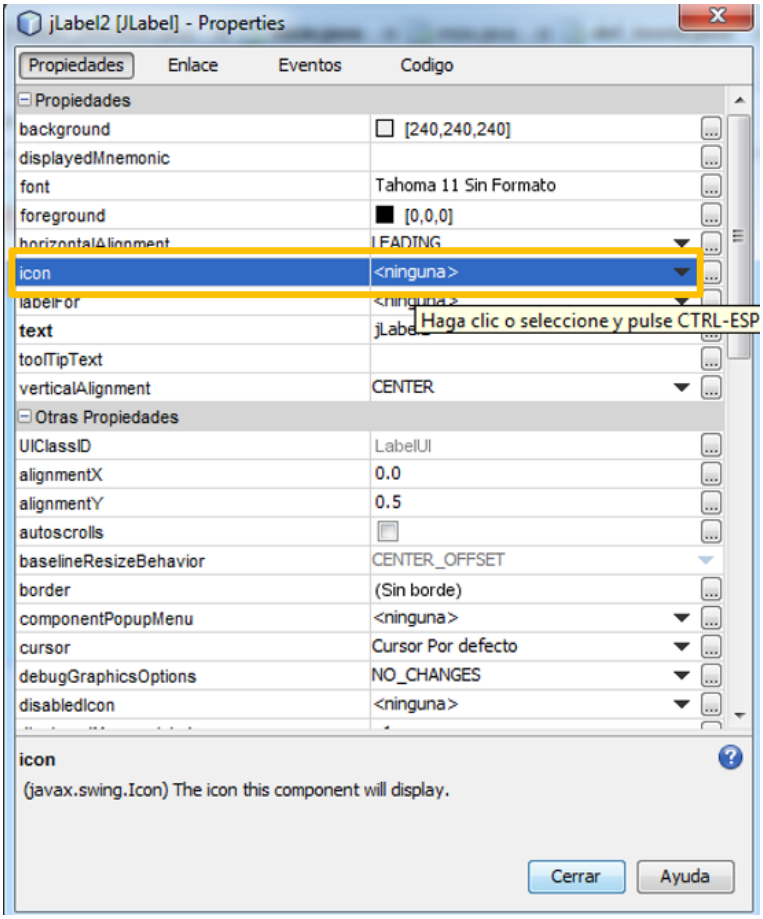
Luego añadimos un ETIQUETA Jlabel que nos servirá para añadir el título ***TEORÍA DE ERRORES.***

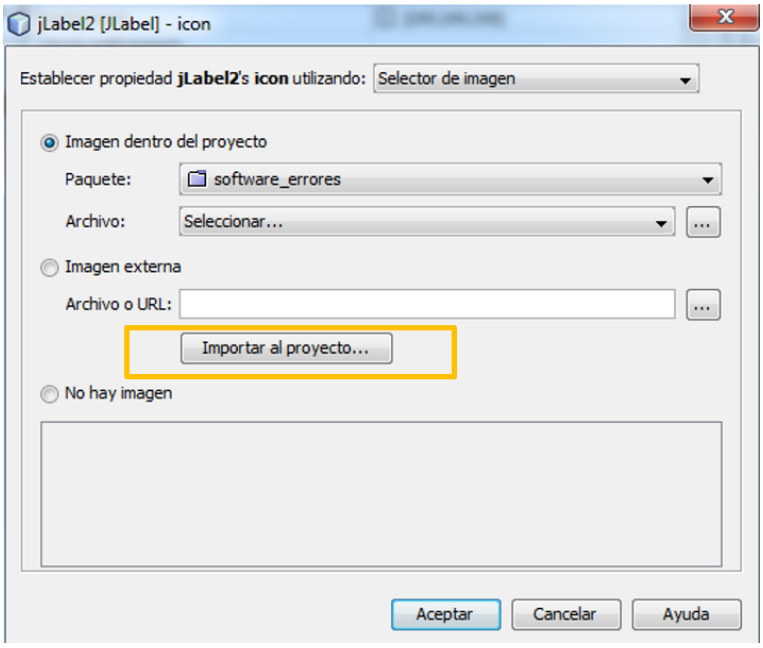




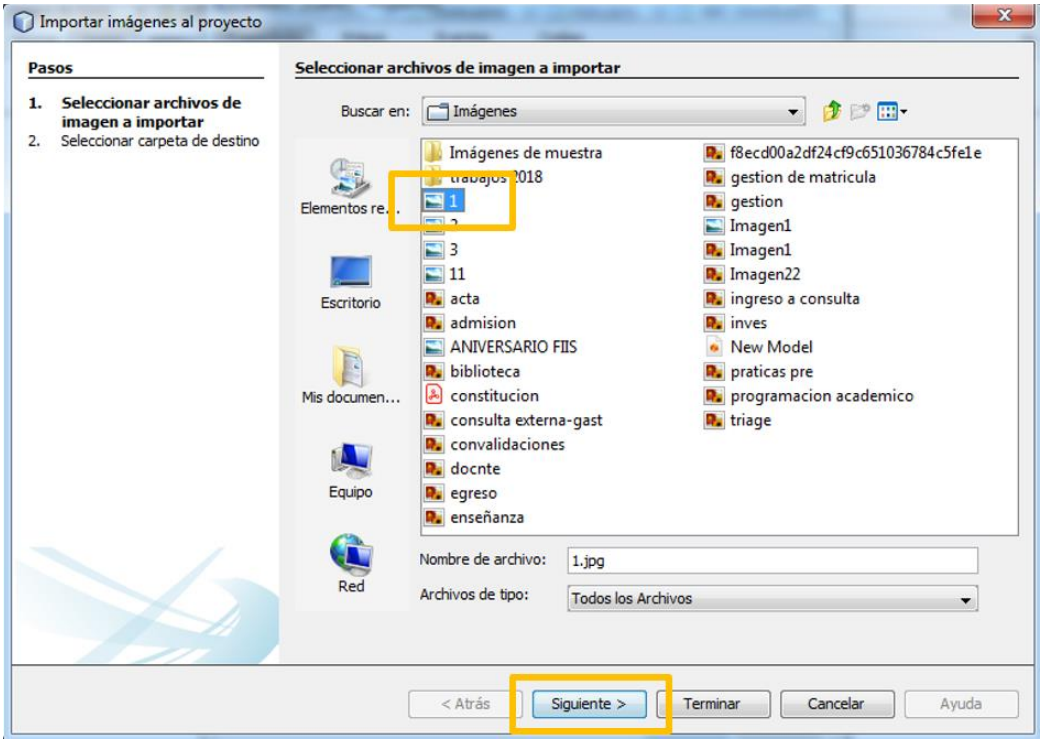
Para añadir una imagen dentro de la interface realizamos las siguientes operaciones. ***INSERTAMOS UNA ETIQUETA > CLIC DERECHO > PROPIEDADES > ICON > IMPORTAR AL PROYECTO.***


SOFTWARE_ERRORES - NetBeans IDE 8.2
Archivo Editar Ver Navegar Fuente Reestructurar Ejecutar Depurar Profile Team Herramientas Ventana Ayuda
<default config>
Proyectos
SOFTWARE_ERRORES
SOFTWARE_ERRORES
Paquetes de fuentes
software_errores
SOFTWARE_ERRORES.java
def_teoria.java
inicio.java
Bibliotecas
Página de Inicio
SOFTWARE_ERRORES.java
inicio.java
Source
Design
History
Utilice el botón Fuente (en la barra de herramientas) para cambiar al código fuente.
TEORÍA DE ERRORES
jLabel2

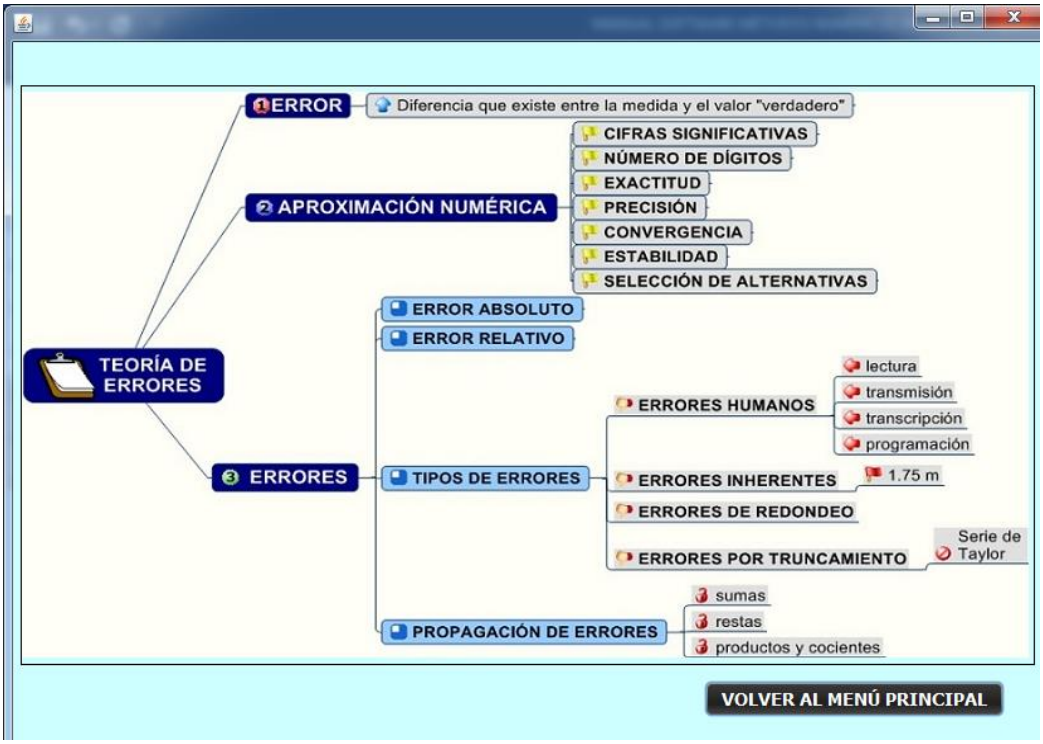

jLabel2 [JLabel] - Properties
Propiedades
Enlace
Eventos
Codigo
Propiedades
background
[240,240,240]
displayedMnemonic
font
Tahoma 11 Sin Formato
foreground
[0,0,0]
icon
<ninguna>
Haga clic o seleccione y pulse CTRL-ESP
text
verticalAlignment
CENTER
Otras Propiedades
UIClassID
LabelUI
alignmentX
0.0
alignmentY
0.5
autoscrolls
baselineResizeBehavior
CENTER_OFFSET
border
(Sin borde)
componentPopupMenu
<ninguna>
cursor
Cursor Por defecto
debugGraphicsOptions
NO_CHANGES
disabledIcon
<ninguna>
icon
(javax.swing.Icon) The icon this component will display.
Cerrar
Ayuda



Importamos una imagen a la ventana, y tendremos nuestra interface terminada.



Además, añadimos un botón (jButton) al que llamaremos ***VOLVER MENÚ PRINCIPAL***, que nos servirá para volver a la ventana de inicio. Y finalmente tenemos la ventana teoría de errores.



Los siguiente JFrame para las otras definiciones serán:

def_errorabsyrel

def_errorred

def_relpor

def_sertaymac

método

principal

A continuación, mostramos cada uno de las codificaciones de las ventanas de la primera opción ***DEFINICIONES***.

a) def_teoría

```
1   /*
2    * To change this license header, choose License Headers in Project Properties.
3    * To change this template file, choose Tools | Templates
4    * and open the template in the editor.
5    */
6   package software_errores;
7
8   /**
9    *
10   * @author 4CER
11   */
12  public class def_teoria extends javax.swing.JFrame {
13
14      /**
15       * Creates new form def_teoria
16       */
17      public def_teoria() {
18          initComponents();
19      }
20
21      /**
22       * This method is called from within the constructor to initialize the form.
23       * WARNING: Do NOT modify this code. The content of this method is always
24       * regenerated by the Form Editor.
25       */
26      @SuppressWarnings("unchecked")
27      Generated Code
78
```

```
79      /**
80       * @param args the command line arguments
81       */
82      public static void main(String args[]) {
83          /* Set the Nimbus look and feel */
84          Look and feel setting code (optional)
105
106         /* Create and display the form */
            java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
109                 new def_teoria().setVisible(true);
110             }
111         });
112     }
113
114     // Variables declaration - do not modify
115     private javax.swing.JLabel jLabel1;
116     private javax.swing.JLabel jLabel2;
117     private javax.swing.JPanel jPanel1;
118     // End of variables declaration
119 }
```

b) def_errorabsyrel


```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package software_errores;

/**
 *
 * @author 4CER
 */
public class def_errorabsyrel extends javax.swing.JFrame {

    /**
     * Creates new form def_errorabsyrel
     */
    public def_errorabsyrel() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
22      /**
23       * This method is called from within the constructor to initialize the form.
24       * WARNING: Do NOT modify this code. The content of this method is always
25       * regenerated by the Form Editor.
26       */
27      @SuppressWarnings("unchecked")
28      Generated Code
115
116     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
117         // TODO add your handling code here:
118         new inicio().setVisible(true);
119         this.setVisible(false);
120     }
121
122     /**
123      * @param args the command line arguments
124      */
125     public static void main(String args[]) {
126         /* Set the Nimbus look and feel */
127         Look and feel setting code (optional)
148
149         /* Create and display the form */
            java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
152                 new def_errorabsyrel().setVisible(true);
153             }
154         });
155     }
156
157     // Variables declaration - do not modify
158     private javax.swing.JButton jButton1;
159     private javax.swing.JLabel jLabel1;
160     private javax.swing.JLabel jLabel2;
161     private javax.swing.JLabel jLabel3;
162     private javax.swing.JLabel jLabel4;
163     private javax.swing.JPanel jPanel1;
164     // End of variables declaration
```

c) def_relpor

ERROR RELATIVO PORCENTUAL

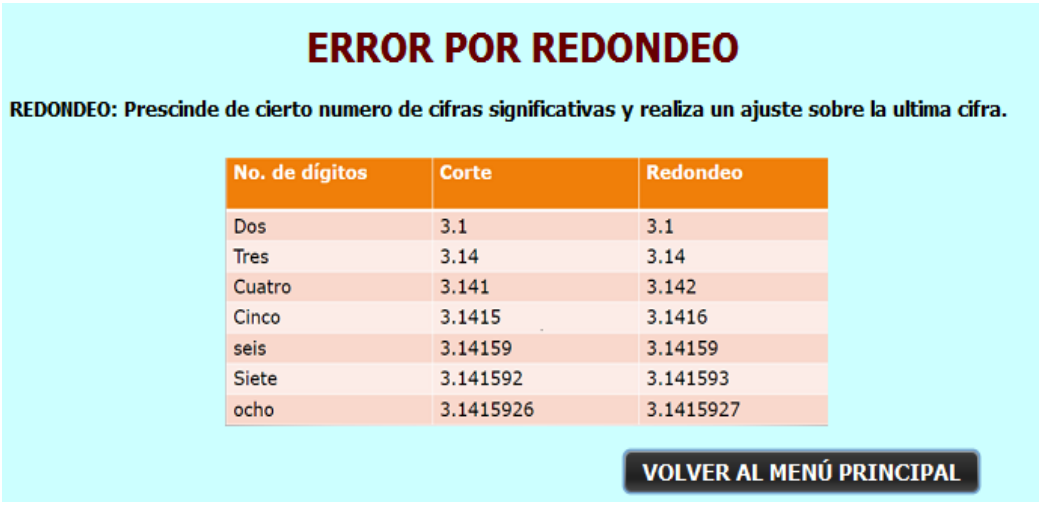
Suele ser un mejor indicador de la precisión, es más independiente de la escala usada, y esto es una propiedad más que deseable.

$$Er = \frac{\varepsilon}{a} = \frac{\tilde{a} - a}{a} = \frac{Error}{Valor\ Verdadero} * 100$$

VOLVER AL MENÚ PRINCIPAL

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package software_errores;

/**
 *
 * @author 4CER
 */
public class def_relpor extends javax.swing.JFrame {

    /**
     * Creates new form def_relpor
     */
    public def_relpor() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new inicio().setVisible(true);
        this.setVisible(false);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new def_relpor().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JPanel jPanel1;
    // End of variables declaration
}
```

d) def_errorred



ERROR POR REDONDEO

REDONDEO: Prescinde de cierto numero de cifras significativas y realiza un ajuste sobre la ultima cifra.

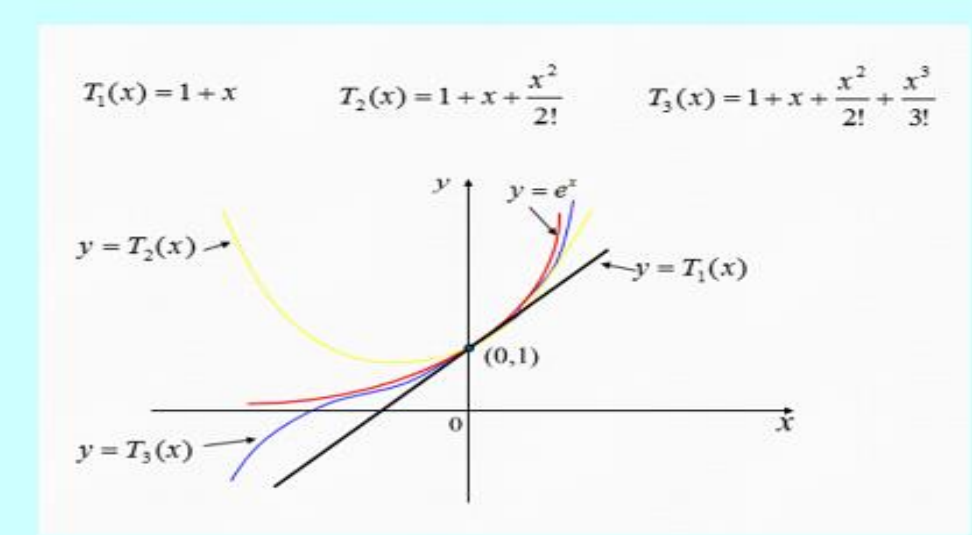
No. de dígitos	Corte	Redondeo
Dos	3.1	3.1
Tres	3.14	3.14
Cuatro	3.141	3.142
Cinco	3.1415	3.1416
seis	3.14159	3.14159
Siete	3.141592	3.141593
ocho	3.1415926	3.1415927

VOLVER AL MENÚ PRINCIPAL

```
1   /*
2    * To change this license header, choose License Headers in Project Properties.
3    * To change this template file, choose Tools | Templates
4    * and open the template in the editor.
5    */
6   package software_errores;
7
8   /**
9    *
10   * @author 4CER
11   */
12  public class def_errorred extends javax.swing.JFrame {
13
14      /**
15       * Creates new form def_errorred
16       */
17      public def_errorred() {
18          initComponents();
19          this.setLocationRelativeTo(null);
20      }
21
22      /**
23       * This method is called from within the constructor to initialize the form.
24       * WARNING: Do NOT modify this code. The content of this method is always
25       * regenerated by the Form Editor.
26       */
27      @SuppressWarnings("unchecked")
28      Generated Code
110
111     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
110
111     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
112         // TODO add your handling code here:
113          new inicio().setVisible(true);
114         this.setVisible(false);
115     }
116
117     /**
118      * @param args the command line arguments
119      */
120     public static void main(String args[]) {
121         /* Set the Nimbus look and feel */
122          Look and feel setting code (optional)
143
144         /* Create and display the form */
            java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
147                 new def_errorred().setVisible(true);
148             }
149         });
150     }
151
152     // Variables declaration - do not modify
153     private javax.swing.JButton jButton1;
154     private javax.swing.JLabel jLabel1;
155     private javax.swing.JLabel jLabel2;
156     private javax.swing.JLabel jLabel4;
157     private javax.swing.JPanel jPanel1;
158     // End of variables declaration
159 }
```

e) def_sertaymac

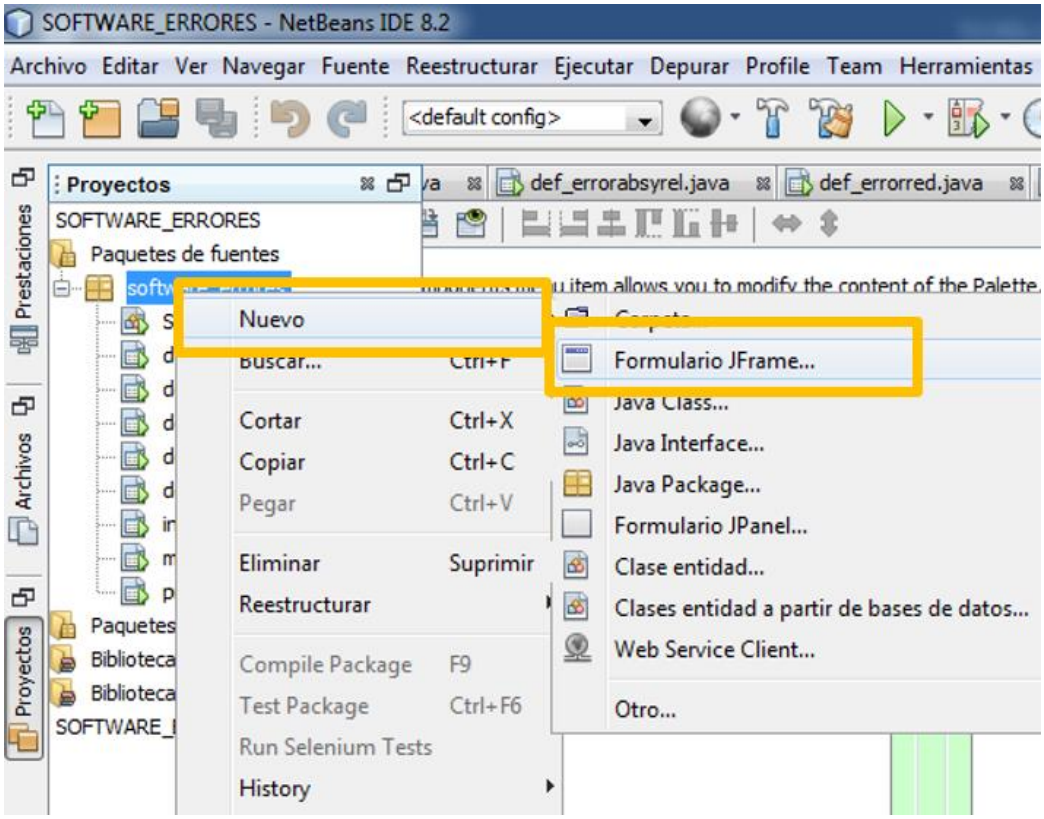


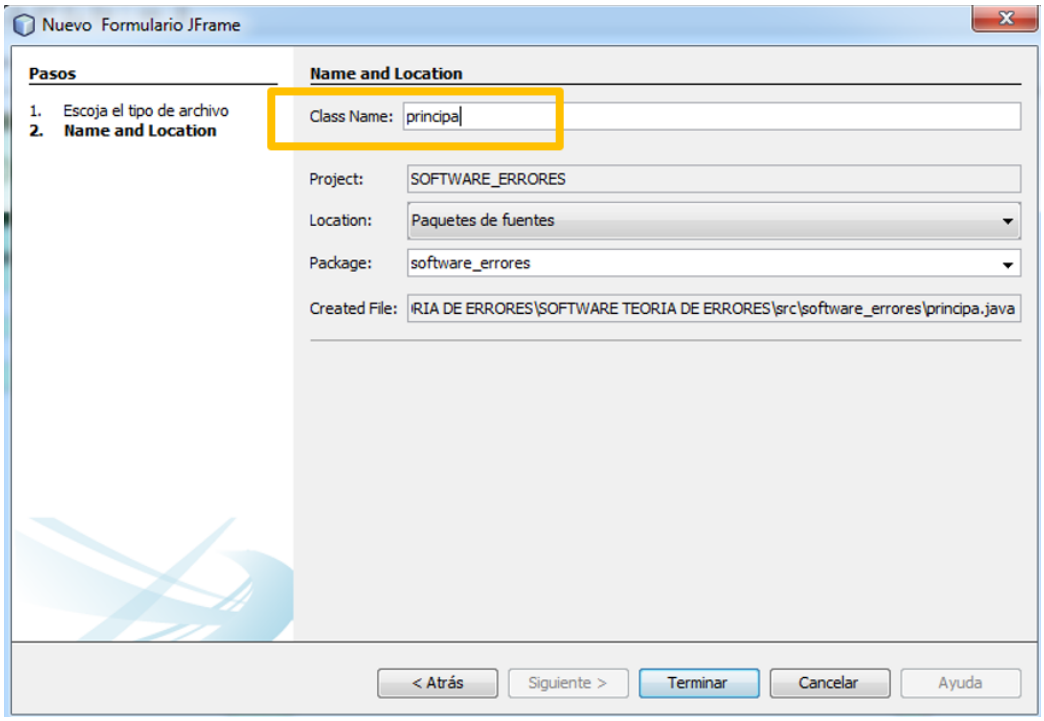

```
1   /*
2    * To change this license header, choose License Headers in Project Properties.
3    * To change this template file, choose Tools | Templates
4    * and open the template in the editor.
5    */
6   package software_errores;
7
8   /**
9    *
10   * @author 4CER
11   */
12  public class def_sertaymac extends javax.swing.JFrame {
13
14      /**
15       * Creates new form def_sertaymac
16       */
17      public def_sertaymac() {
18          initComponents();
19          this.setLocationRelativeTo(null);
20      }
21
22      /**
23       * This method is called from within the constructor to initialize the form.
24       * WARNING: Do NOT modify this code. The content of this method is always
25       * regenerated by the Form Editor.
26       */
27      @SuppressWarnings("unchecked")
28      Generated Code
134
```

```
135     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
136         // TODO add your handling code here:
137        new inicio().setVisible(true);
138         this.setVisible(false);
139     }
140
141     /**
142      * @param args the command line arguments
143      */
144     public static void main(String args[]) {
145         /* Set the Nimbus look and feel */
146          Look and feel setting code (optional)
167
168         /* Create and display the form */
            java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
171                 new def_sertaymac().setVisible(true);
172             }
173         });
174     }
175     // Variables declaration - do not modify
176     private javax.swing.JButton jButton1;
177     private javax.swing.JLabel jLabel1;
178     private javax.swing.JLabel jLabel2;
179     private javax.swing.JLabel jLabel3;
180     private javax.swing.JLabel jLabel4;
181     private javax.swing.JLabel jLabel5;
182     private javax.swing.JLabel jLabel6;
183     private javax.swing.JPanel jPanel1;
184     // End of variables declaration
```

En la segunda parte de nuestra ventana de inicio, vamos a insertar un Jframe para el botón ***TIPOS DE ERRORES*** y otro Jframe para el botón ***MÉTODOS ITERATIVOS***.

Para añadir un Jframe al botón ***TIPOS DE ERRORES*** realizamos la siguiente operación. ***SOFTWARE_ERRORES > NUEVO > FORMULARIO JFRAME***. Y llamamos a nuestra nueva ventana ***PRINCIPAL > TERMINAR***.





Insertamos un JPanel que nos servirá como el fondo de nuestra ventana, cambiaremos el color del fondo de nuestra ventana en color verde, e insertamos una ETIQUETA (JLabel) y editaremos el título ***MÉTODOS NUMÉRICOS-TIPOS DE ERRORES***. De igual manera añadimos ***6 ETIQUETAS (JLabel)*** que nos servirá como los títulos de nuestras dos opciones que tendremos en nuestra ventana.

Error absoluto, relativo y relativo porcentual

Valor real

Valor aproximado

Error por truncamiento y redondeo

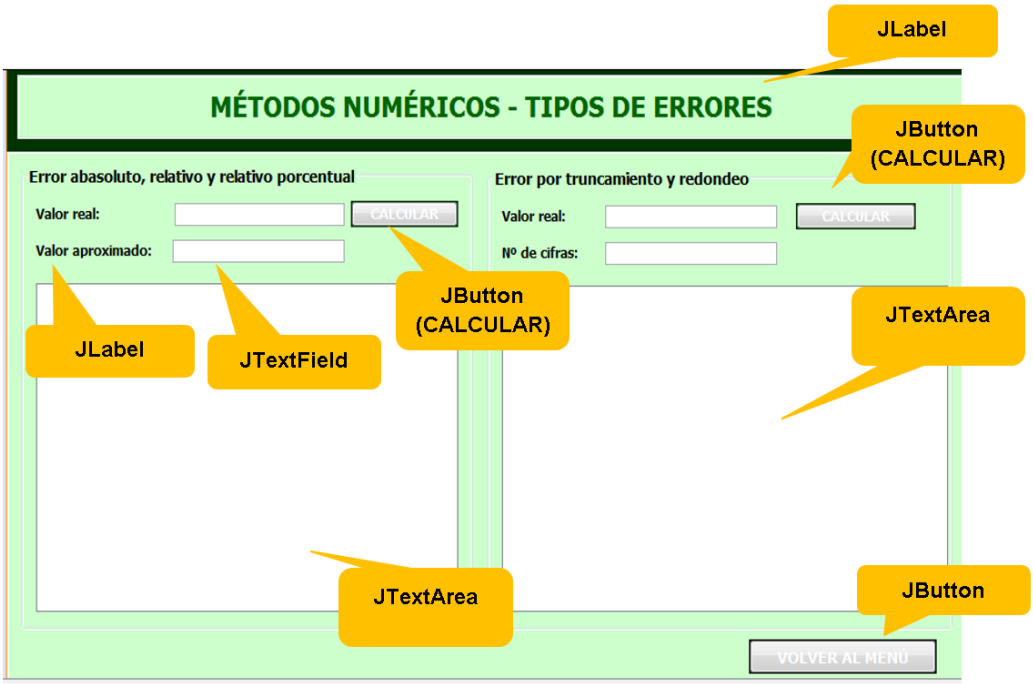
Valor real

N° de cifras

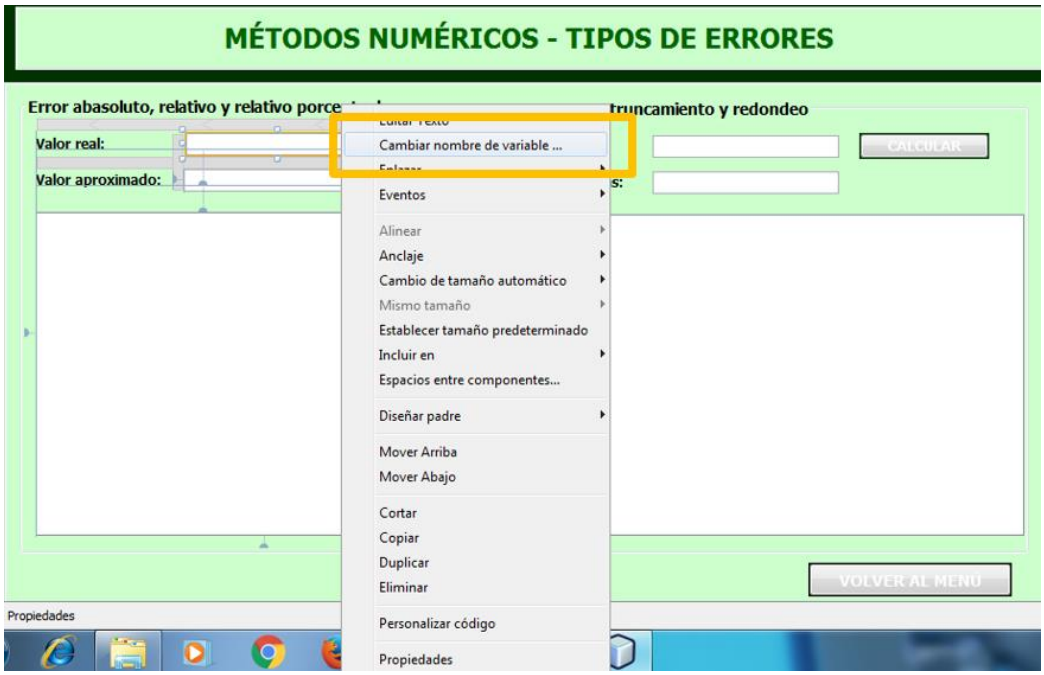
Añadimos 4 campos de texto que irán al lado de nuestras ***ETIQUETAS (VALOR REAL, VALOR APROXIMADO – VALOR REAL, N° DE CIFRAS)***

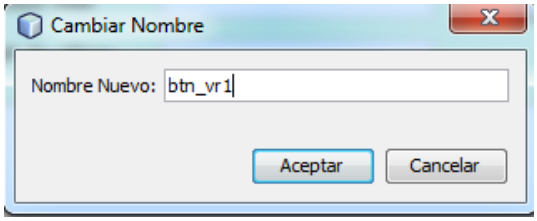
Luego añadimos 3 botones (JButton) a las cuales llamaremos ***(CALCULAR – VOLVER A MENÚ)***

Por último, añadiremos 2 JTextArea que nos servirá para la ejecución de nuestro programa, en esta área saldrá la respuesta de nuestra ventana.

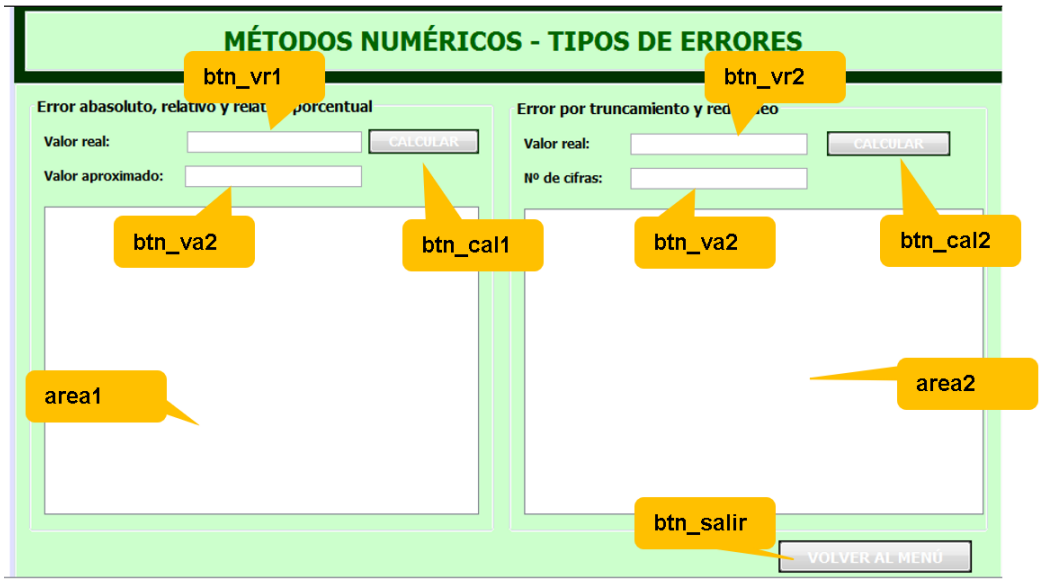


Cambiamos los nombres de las variables de algunos de nuestros componentes para poder codificar posteriormente. Dando clic derecho sobre el componente.





Cambiamos el nombre de las siguientes variables para la posterior codificación.



La codificación de esta ventana es la siguiente, con ello funcionará dicha ventana.

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package software_errores;

/**
 *
 * @author MARSHMELO
 */
public class principal extends javax.swing.JFrame {

    /**
     * Creates new form principal
     */
    public principal() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
```

```java
        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jLabel5 = new javax.swing.JLabel();
        jPanel3 = new javax.swing.JPanel();
        jPanel4 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        btn_vr1 = new javax.swing.JTextField();
        btn_cal1 = new javax.swing.JButton();
        jLabel2 = new javax.swing.JLabel();
        btn_va1 = new javax.swing.JTextField();
        jScrollPane1 = new javax.swing.JScrollPane();
        area1 = new javax.swing.JTextArea();
        jPanel5 = new javax.swing.JPanel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        btn_vr2 = new javax.swing.JTextField();
        btn_va2 = new javax.swing.JTextField();
        btn_cal2 = new javax.swing.JButton();
        jScrollPane2 = new javax.swing.JScrollPane();
        area2 = new javax.swing.JTextArea();
        btn_salir = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jPanel1.setBackground(new java.awt.Color(0, 51, 0));

        jPanel2.setBackground(new java.awt.Color(204, 255, 204));
        jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(""));
```

```
        jLabel5.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N
        jLabel5.setForeground(new java.awt.Color(0, 102, 0));
        jLabel5.setText("MÉTODOS NUMÉRICOS - TIPOS DE ERRORES");

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(183, 183, 183)
                .addComponent(jLabel5)
                .addContainerGap(418, Short.MAX_VALUE))
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel5)
                .addContainerGap())
        );

        jPanel3.setBackground(new java.awt.Color(204, 255, 204));
        jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(""));

        jPanel4.setBackground(new java.awt.Color(204, 255, 204));
        jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Error abasoluto, relativo y
        jPanel4.setForeground(new java.awt.Color(255, 255, 255));

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
```

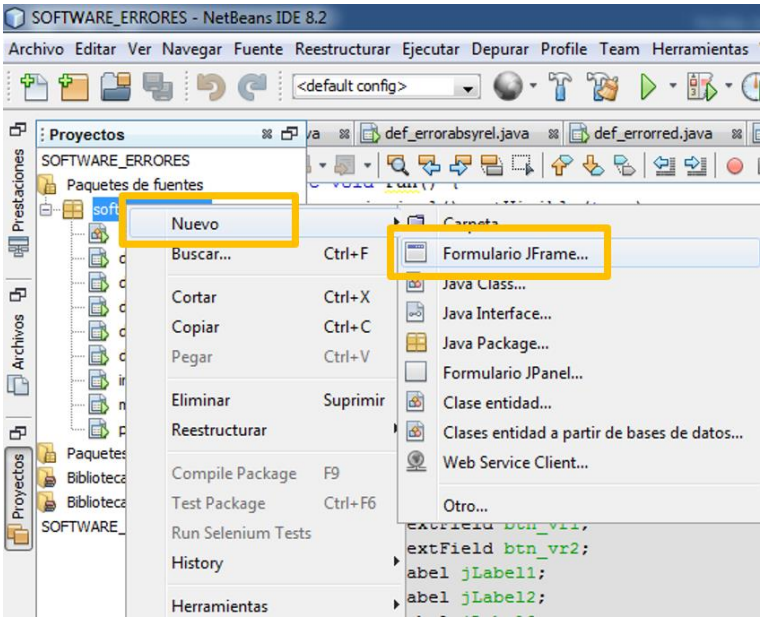
```
        btn_cal1.setBackground(new java.awt.Color(0, 51, 0));
        btn_cal1.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        btn_cal1.setForeground(new java.awt.Color(255, 255, 255));
        btn_cal1.setText("CALCULAR");
        btn_cal1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btn_cal1ActionPerformed(evt);
            }
        });

        jLabel2.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        jLabel2.setText("Valor aproximado:");

        area1.setColumns(20);
        area1.setRows(5);
        jScrollPane1.setViewportView(area1);

        javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
        jPanel4.setLayout(jPanel4Layout);
        jPanel4Layout.setHorizontalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jScrollPane1)
                    .addGroup(jPanel4Layout.createSequentialGroup()
                        .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 95, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 121, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

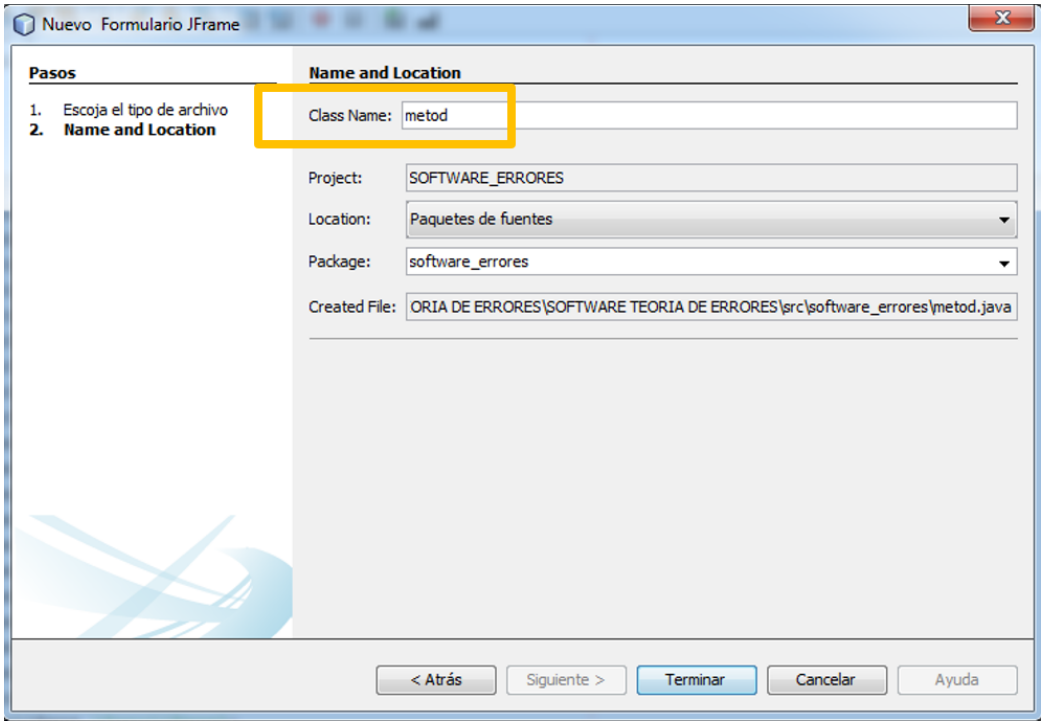
```
270     private void btn_cal2ActionPerformed(java.awt.event.ActionEvent evt) {
            double x=0, r_et=0, r_er=0, t=0, r=0, r_et2=0, a;
            int p=0, y=0, l=0;
273         x=Double.parseDouble(btn_vr2.getText());
274         y=Integer.parseInt(btn_va2.getText());
275         l=(btn_vr2.getText()).length();
276         p=(btn_vr2.getText()).indexOf('.');
277         r_et=(x-Math.floor(x))*Math.pow(10,(y-p));
278         r_et=Math.round(r_et);
279         r_et2=Math.floor((x-Math.floor(x))*Math.pow(10,(y-p)));
            r_et=(r_et/Math.pow(10,(y-p)))+Math.floor(x);
281         r_et2=(r_et2/Math.pow(10,(y-p)))+Math.floor(x);
282         t=x-r_et2;
283
284         a=t*Math.pow(10,(l-p-1));
285         a=Math.round(a);
286         a=a/Math.pow(10,(l-p-1));
287         r=1-a;
288
289         area2.setText("valor = "+x+"\n"+
290                 "n° de cifras = "+y+"\n\n"+
291                 "RESULTADOS: ------------------------------"+"\n\n"+
292                 "ERROR POR TRUNCAMIENTO: "+"\n"+
293                 "Se redondea "+x+" en "+y+" cifras = "+r_et2+"\n"+
294                 "error de truncamiento = "+a+"\n\n"+
295
296                 "ERROR POR REDONDEO: "+"\n"+
297                 "Se redondea "+x+" en "+y+" cifras = "+r_et2+"\n"+
298                 "error de redondeo = "+r+"\n");
```

```
301
302     private void btn_salirActionPerformed(java.awt.event.ActionEvent evt) {
303         new inicio().setVisible(true);
304         this.setVisible(false);
305     }
306
307     private void btn_cal1ActionPerformed(java.awt.event.ActionEvent evt) {
            float x=0, y=0, r_ea=0, r_er=0, r_erp=0;
309         x=Float.parseFloat(btn_vr1.getText());
310         y=Float.parseFloat(btn_va1.getText());
311
312         r_ea=Math.abs(x-y);
313         r_er=Math.abs((x-y)/(x));
314         r_erp=(Math.abs((x-y)/(x)))*100;
315
316         area1.setText("valor real = "+Float.toString(x)+"\n"+
317             "valor absoluto = "+Float.toString(y)+"\n\n"+
318             "RESULTADOS: ------------------------------"+"\n\n"+
319             "ERROR ABSOLUTO: "+"\n"+
320             "= |x-y| = "+"|"+x+"-"+y+"| = "+"|"+(x-y)+"|"+"\n"+
321             "= "+Float.toString(r_ea)+"\n\n"+
322
323             "ERROR RELATIVO: "+"\n"+
324             "= |(x-y)/x| = "+"|("+x+"-"+y+")/("+x+")| = "+"|"+((x-y)/x)+"|"+"\n"+
325             "= "+Float.toString(r_er)+"\n\n"+
326
327             "ERROR RELATIVO PORCENTUAL: "+"\n"+
328             "= |(x-y)/x|*(100) = "+"|("+x+"-"+y+")/("+x+")|(100) = "+"|"+((x-y)/x)+"|(100)"+"\n"+
329             "= "+Float.toString(r_erp));
```

```
337     public static void main(String args[]) {
338         /* Set the Nimbus look and feel */
339         Look and feel setting code (optional)
360
361         /* Create and display the form */
            java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
364                 new principal().setVisible(true);
365             }
366         });
367     }
368
369     // Variables declaration - do not modify
370     private javax.swing.JTextArea area1;
371     private javax.swing.JTextArea area2;
372     private javax.swing.JButton btn_cal1;
373     private javax.swing.JButton btn_cal2;
374     private javax.swing.JButton btn_salir;
375     private javax.swing.JTextField btn_va1;
376     private javax.swing.JTextField btn_va2;
377     private javax.swing.JTextField btn_vr1;
378     private javax.swing.JTextField btn_vr2;
379     private javax.swing.JLabel jLabel1;
380     private javax.swing.JLabel jLabel2;
381     private javax.swing.JLabel jLabel3;
382     private javax.swing.JLabel jLabel4;
383     private javax.swing.JLabel jLabel5;
384     private javax.swing.JPanel jPanel1;
385     private javax.swing.JPanel jPanel2;
386     private javax.swing.JPanel jPanel3;
387     private javax.swing.JPanel jPanel4;
388     private javax.swing.JPanel jPanel5;
389     private javax.swing.JScrollPane jScrollPane1;
390     private javax.swing.JScrollPane jScrollPane2;
391     // End of variables declaration
392 }
```

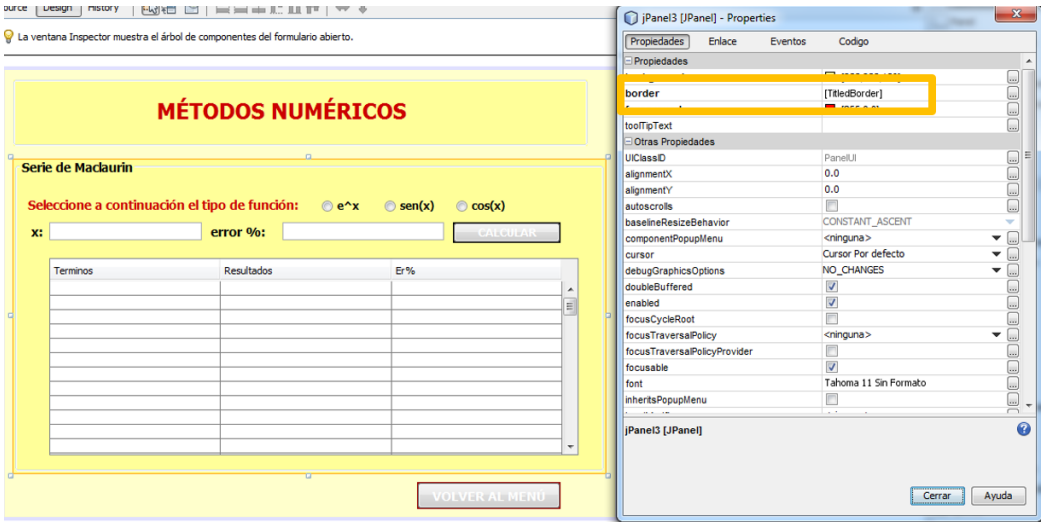
Para la segunda opción de teoría de errores, para el botón ***MÉTODOS ITERATIVOS*** crearemos un nuevo JFrame a la que llamaremos ***METODO***.

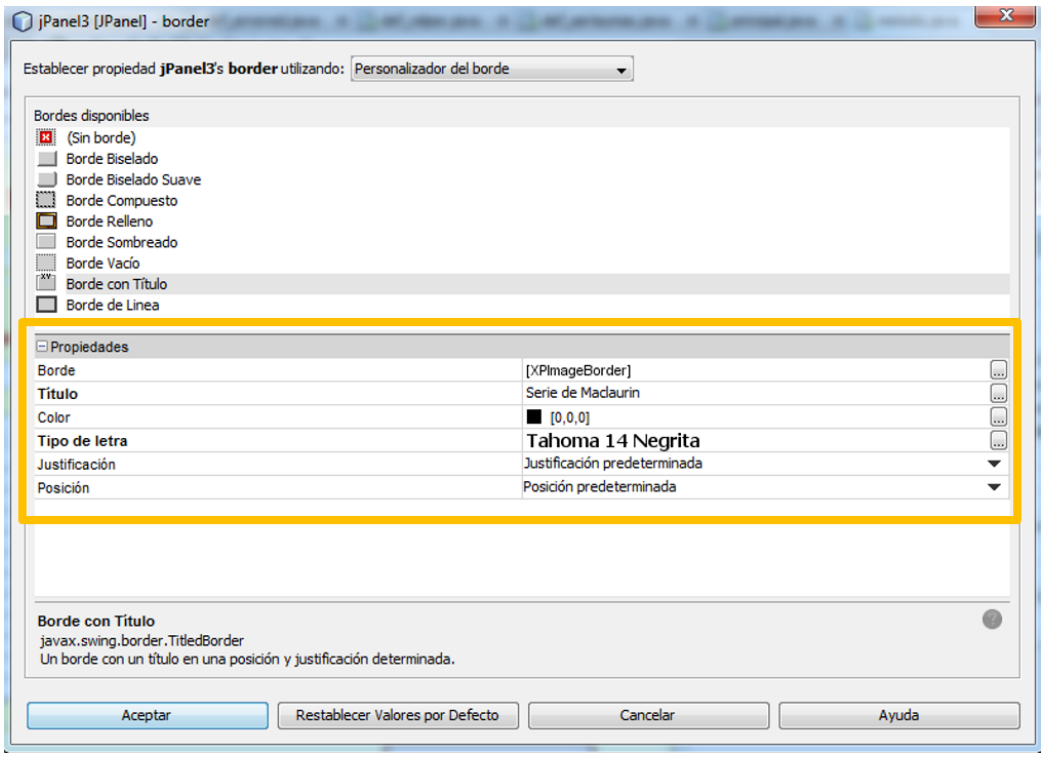




Posteriormente creado el JFrame, en primer lugar añadiremos un JPanel que nos servirá como el fondo de nuestra ventana al cual cambiaremos de color. Luego añadiremos una ETIQUETA (JLabel) que nos servirá como el título de nuestra ventana al cual denominaremos ***MÉTODOS NUMÉRICOS***.

Añadimos un JPanel al que llamaremos Serie de Maclaurin





Luego añadimos ETIQUETAS (JLabel) a las cuales llamaremos:

Seleccione a continuación el tipo de función

X:

Error:

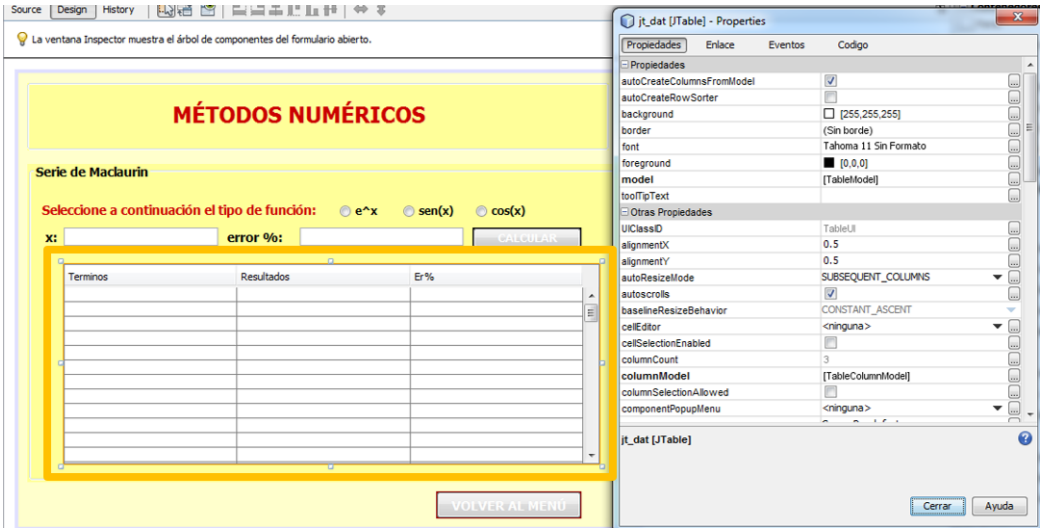
Luego agregaremos 2 JTextField que irán acompañados al ***X: Y ERROR%:*** Posteriormente añadimos un JButton (***CALCULAR***) y otro JButton (***VOLVER AL MENÚ***), agregamos 3 JRadioButton:

e^x

sen(x)

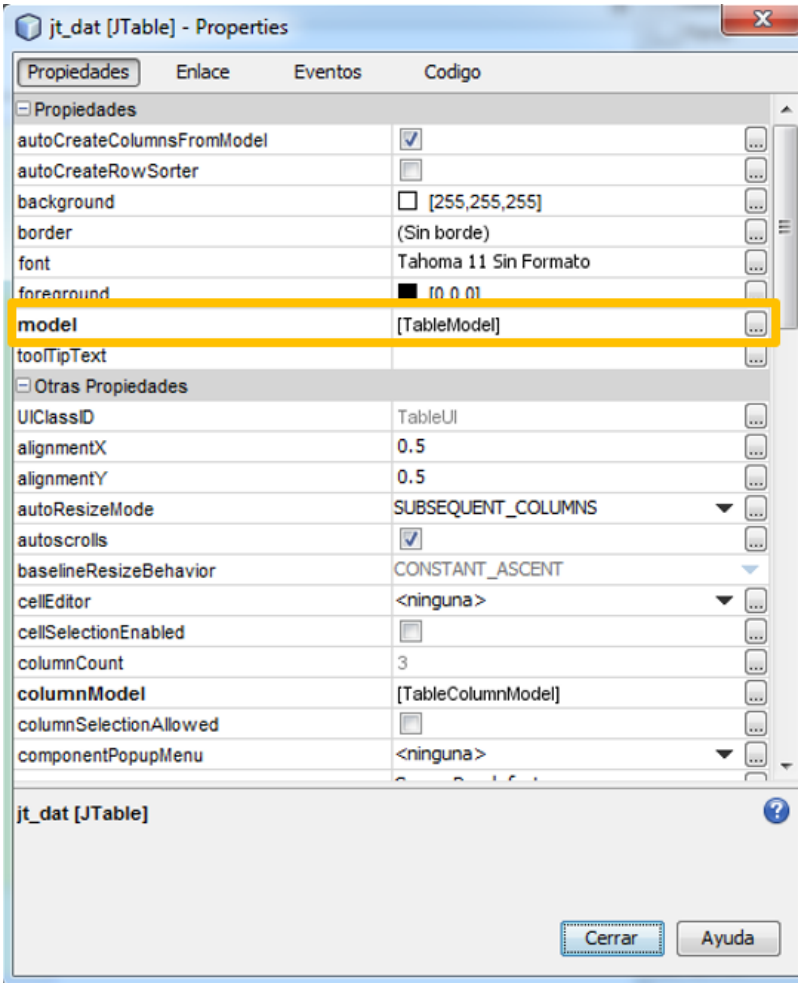
cos(x)

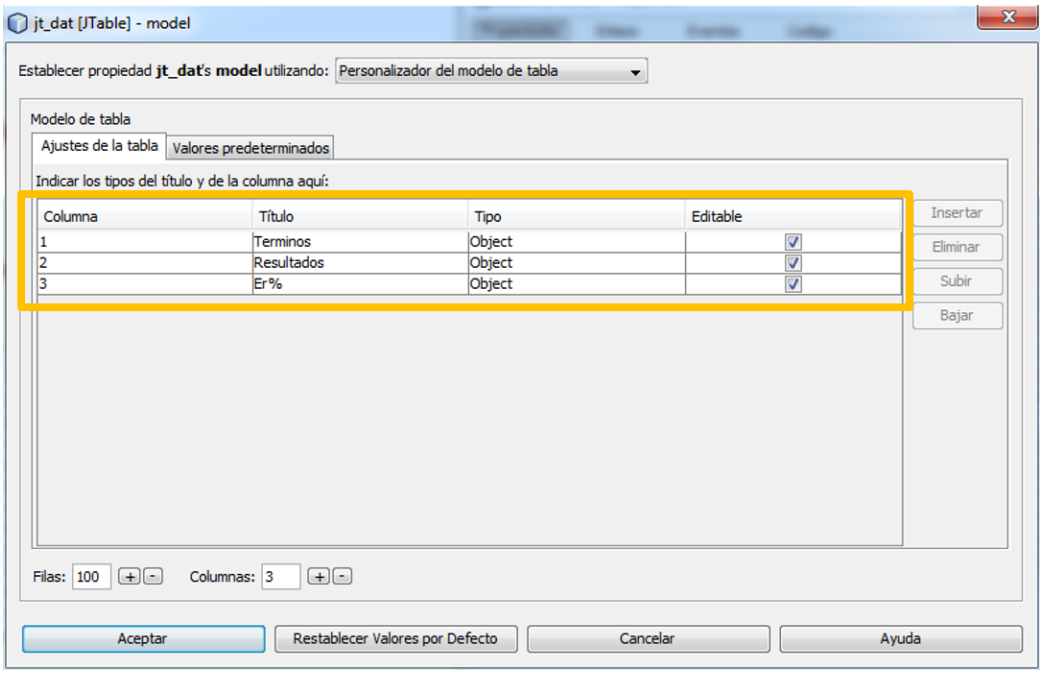
Finalmente añadimos un JTable que nos servirá para calcular los resultados.



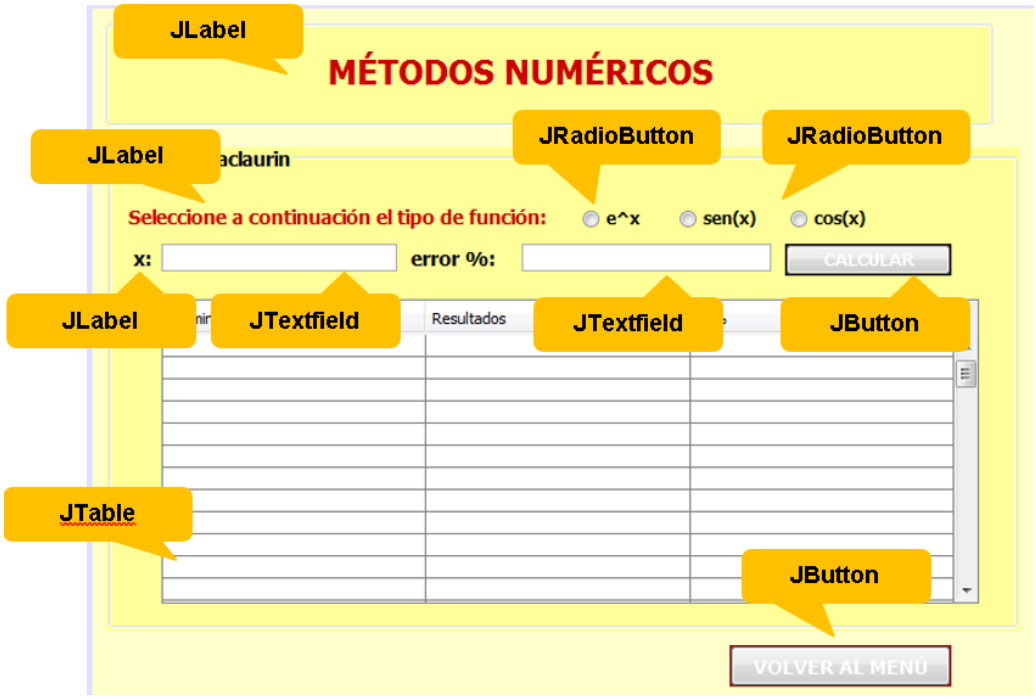
Y modificamos las tres columnas donde se observarán los resultados posteriores.

Para ello damos clic en la opción MODEL y editamos las características del JTable.

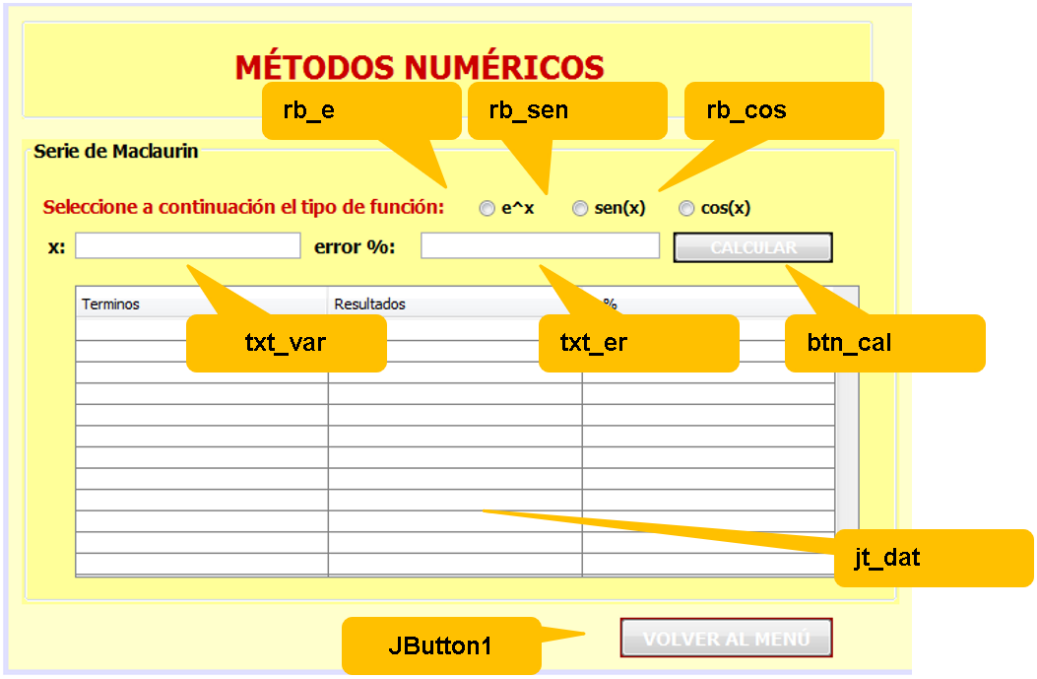




Los componentes añadidos los observaremos a continuación



Luego cambiaremos los nombres de las variables para su posterior codificación.



A continuación, observamos la codificación de la ventana

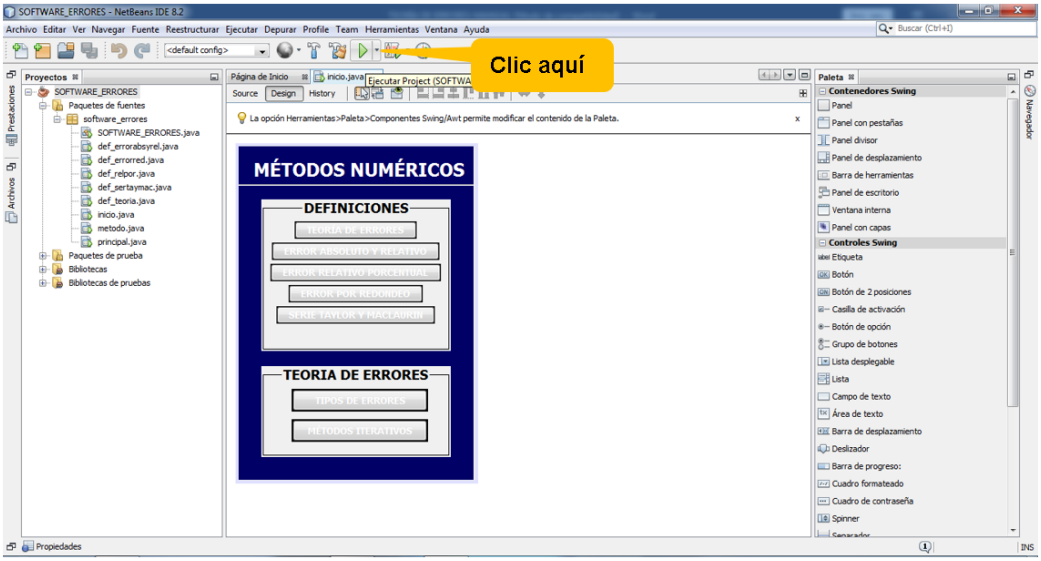
```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package software_errores;

import java.util.Scanner;

/**
 *
 * @author ADVANCE
 */
public class metodo extends javax.swing.JFrame {

    /**
     * Creates new form metodo
     */
    public metodo() {
        initComponents();
        this.setLocationRelativeTo(null);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
```

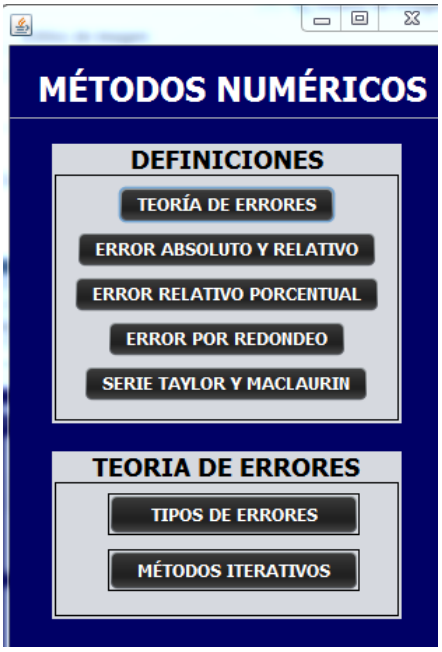
```
249     @SuppressWarnings("empty-statement")
250     private void btn_calActionPerformed(java.awt.event.ActionEvent evt) {
            double x,y,z,res=0,error=0,tem=0,tem2=0;
            int i=0,f=1,j=0,validar=0,w=0,w2=0;
253         x=Double.parseDouble(txt_var.getText());
254         y=Double.parseDouble(txt_er.getText());
255
256         for(int o=0; o<100; o++){
257             jt_dat.setValueAt("", o, 0);
258             jt_dat.setValueAt("", o, 1);
259             jt_dat.setValueAt("", o, 2);
260         }
261
262         if(rb_e.isSelected()){
                i=0;
264             j=0;
265
266             do{
267             i=j;
268             while ( i!=0){
269                 f=f*i;
270                 i--;
271             }
272             res=res+((Math.pow(x,j))/f);
273             error=(Math.abs((res-tem)/res))*100;
274
275             jt_dat.setValueAt(j+1, j, 0);
276             jt_dat.setValueAt(res, j, 1);
277             jt_dat.setValueAt(error, j, 2);
278             i++;
279             j++;
280
281             if((Math.abs(error-tem2))<=y){
282                 validar=1;
283             }
284             tem=res;
285             tem2=error;
286             }while(validar==0);
287         }
288         if(rb_sen.isSelected()){
                i=1;
290             j=1;
291             w=1;
                w2=1;
293             do{
294             i=j;
295
296             while ( i!=0){
297                 f=f*i;
298                 i--;
299             }
300
301             if(w%2==0){
302                 w2=-1;
303             }else{
304                 w2=1;
305             }
306
307             res=res+(w2*(Math.pow(x,j))/f);
308             error=(Math.abs((res-tem)/res))*100;
```

```
303                 }else{
310                     jt_dat.setValueAt(w, w-1, 0);
311                     jt_dat.setValueAt(res, w-1, 1);
312                     jt_dat.setValueAt(error, w-1, 2);
                        i=i+2;
314                     j=j+2;
315                     w++;
316                     if((Math.abs(error-tem2))<=y){
317                         validar=1;
318                     }
319                     tem=res;
320                     tem2=error;
321                     }while(validar==0);
322                 }
323                 if(rb_cos.isSelected()){
                        i=0;
325                     j=0;
326                     w=1;
                        w2=1;
328
329                     do{
330                     i=j;
331
332                     while ( i!=0){
333                         f=f*i;
334                         i--;
335                     }
336
337                     if(w%2==0){
338                         w2=-1;
339                     }else{
340                         w2=1;
341                     }
342
343                     res=res+(w2*(Math.pow(x,j))/f);
344                     error=(Math.abs((res-tem)/res))*100;
345
346                     jt_dat.setValueAt(w, w-1, 0);
347                     jt_dat.setValueAt(res, w-1, 1);
348                     jt_dat.setValueAt(error, w-1, 2);
                        i=i+2;
350                     j=j+2;
351                     w++;
352                     if((Math.abs(error-tem2))<=y){
353                         validar=1;
354                     }
355                     tem=res;
356                     tem2=error;
357                     }while(validar==0);
358                 }
359
360             }
361
362
363             private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
364                 new inicio().setVisible(true);
365                 this.setVisible(false);
366             }
367
368             /**
```

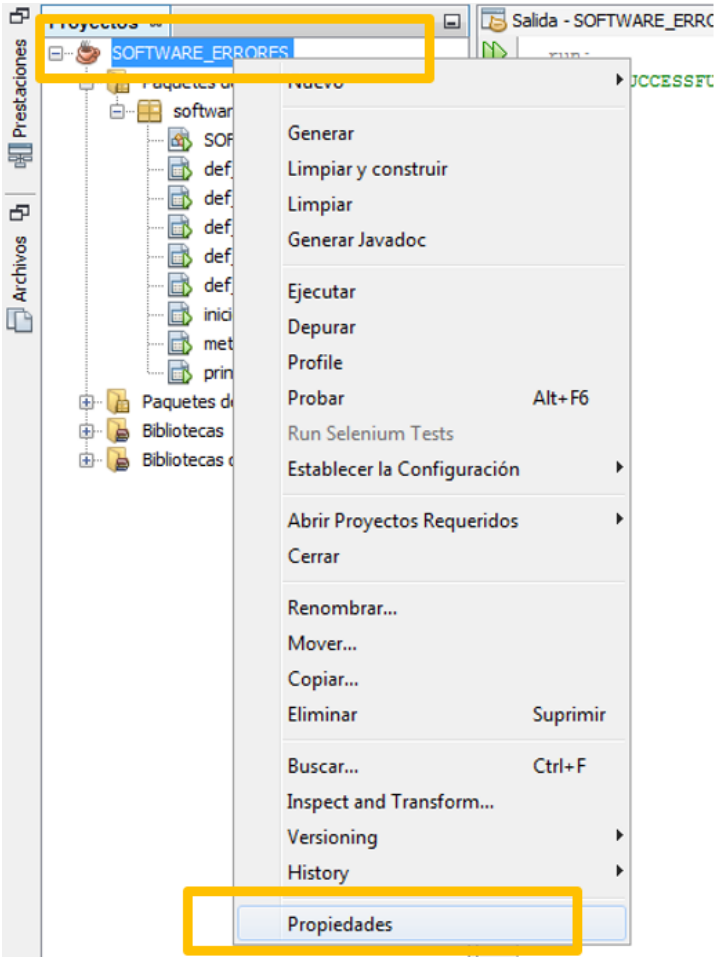
```
368     /**
369      * @param args the command line arguments
370      */
371     public static void main(String args[]) {
372         /* Set the Nimbus look and feel */
373         Look and feel setting code (optional)
394
395         /* Create and display the form */
            java.awt.EventQueue.invokeLater(new Runnable() {
                public void run() {
398                 new metodo().setVisible(true);
399             }
400         });
401     }
402
403     // Variables declaration - do not modify
404     private javax.swing.JButton btn_cal;
405     private javax.swing.ButtonGroup buttonGroup1;
406     private javax.swing.JButton jButton1;
407     private javax.swing.JLabel jLabel1;
408     private javax.swing.JLabel jLabel2;
409     private javax.swing.JLabel jLabel3;
410     private javax.swing.JLabel jLabel5;
411     private javax.swing.JPanel jPanel1;
412     private javax.swing.JPanel jPanel2;
413     private javax.swing.JPanel jPanel3;
414     private javax.swing.JScrollPane jScrollPane2;
415     private javax.swing.JTextField jTextField1;
416     private javax.swing.JTable jt_dat;
417     private javax.swing.JRadioButton rb_cos;
418     private javax.swing.JRadioButton rb_e;
419     private javax.swing.JRadioButton rb_sen;
420     private javax.swing.JTextField txt_er;
421     private javax.swing.JTextField txt_var;
422     // End of variables declaration
423
424
425     }
```

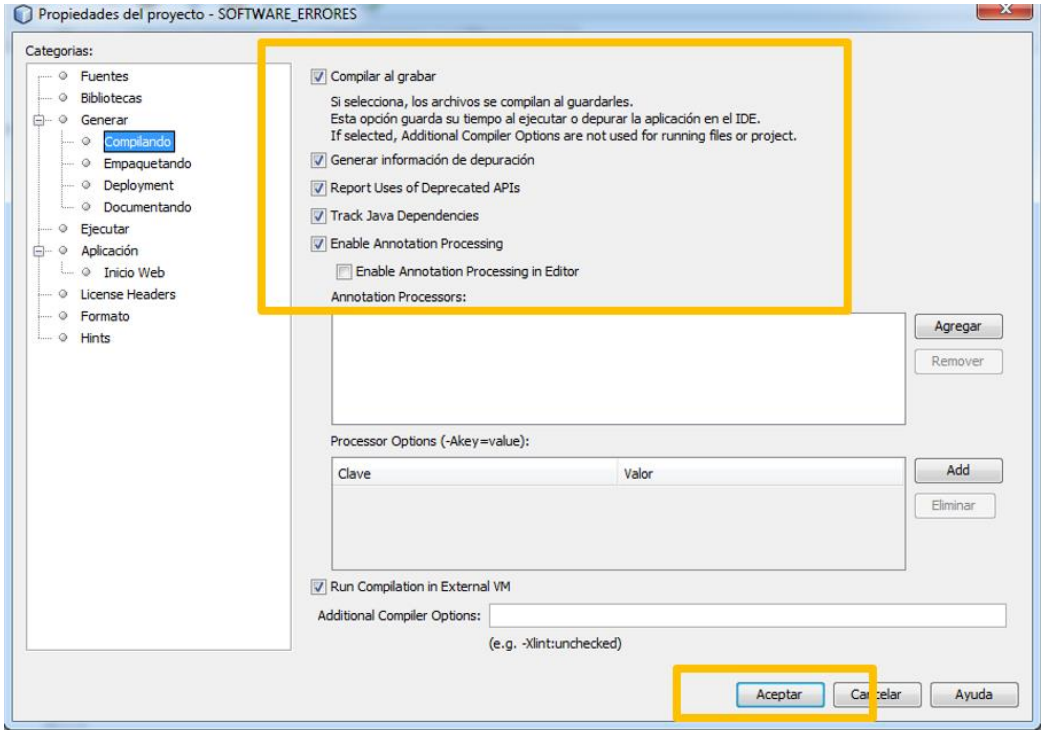
Teniendo en cuenta toda la codificación anterior, procedemos a la ejecución del proyecto.

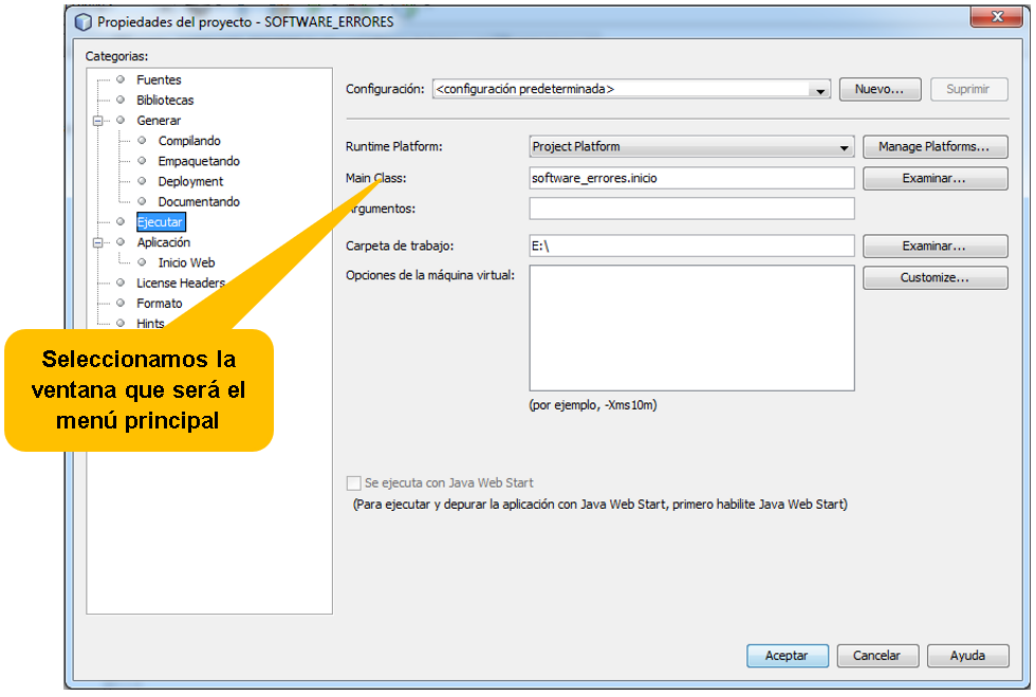


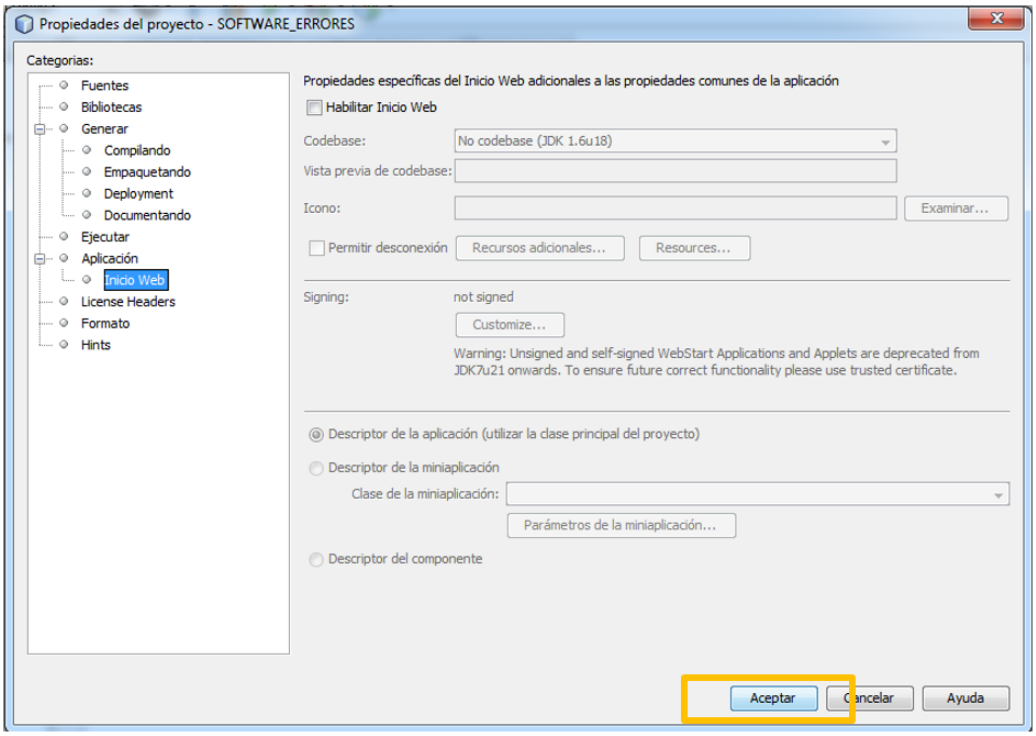


Para crear un archivo jar, realizamos la siguiente operación. CLIC DERECHO SOFTWARE_ERRORES > PROPIEDADES > COMPILANDO > EJECUTAR > INICIO WEB > ACEPTAR.

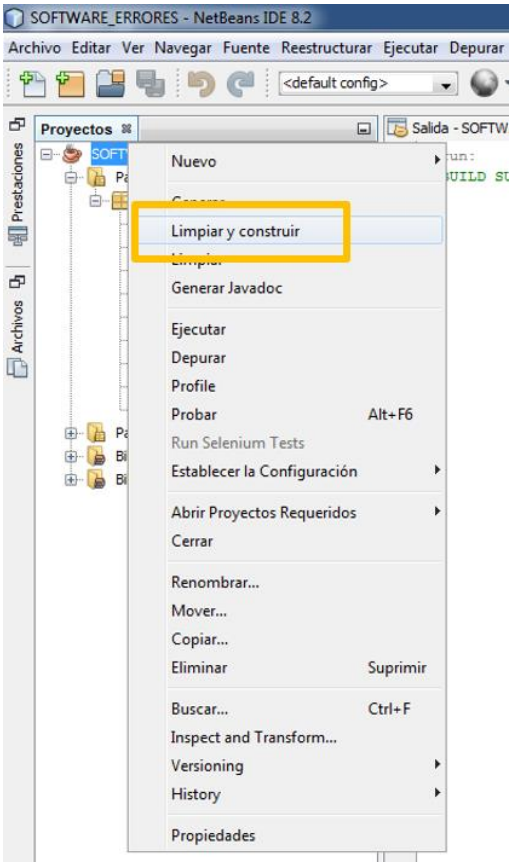


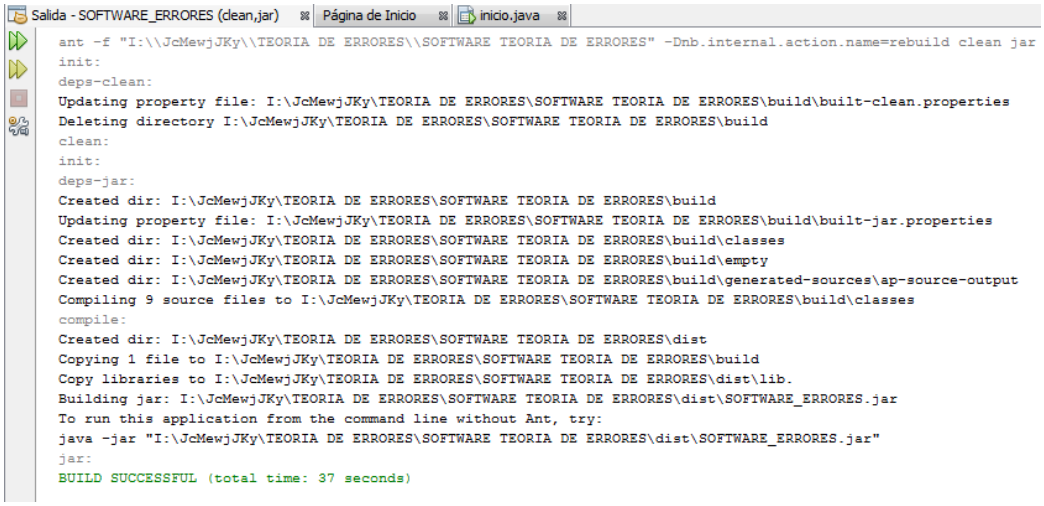

Propiedades del proyecto - SOFTWARE_ERRORES
Categorias:
Fuentes
Bibliotecas
Generar
Compilando
Empaquetando
Deployment
Documentando
Ejecutar
Aplicación
Inicio Web
License Headers
Formato
Hints
Compilar al grabar
Si selecciona, los archivos se compilan al guardarles.
Esta opción guarda su tiempo al ejecutar o depurar la aplicación en el IDE.
If selected, Additional Compiler Options are not used for running files or project.
Generar información de depuración
Report Uses of Deprecated APIs
Track Java Dependencies
Enable Annotation Processing
Enable Annotation Processing in Editor
Annotation Processors:
Agregar
Remover
Processor Options (-Akey=value):
Clave
Valor
Add
Eliminar
Run Compilation in External VM
Additional Compiler Options:
(e.g. -Xlint:unchecked)
Aceptar
Ayuda


Propiedades del proyecto - SOFTWARE_ERRORES
Categorias:
Fuentes
Bibliotecas
Generar
Compilando
Empaquetando
Deployment
Documentando
Ejecutar
Aplicación
Inicio Web
License Headers
Formato
Hints
Configuración: <configuración predeterminada>
Nuevo...
Suprimir
Runtime Platform:
Project Platform
Manage Platforms...
Main Class:
software_errores.inicio
Examinar...
Argumentos:
Carpeta de trabajo:
E:\
Examinar...
Opciones de la máquina virtual:
Customize...
(por ejemplo, -Xms10m)
Se ejecuta con Java Web Start
(Para ejecutar y depurar la aplicación con Java Web Start, primero habilite Java Web Start)
Aceptar
Cancelar
Ayuda
Seleccionamos la ventana que será el menú principal

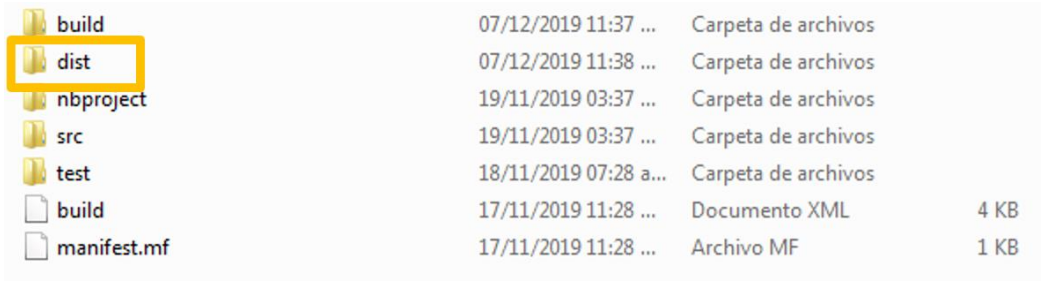


Luego damos clic derecho a SOFTWARE_ERRORES Y seleccionamos la opción LIMPIAR Y CONSTRUIR y dejamos que ejecute. Y automáticamente ya tendremos el archivo jar, dentro del archivo del proyecto, en la carpeta DIST.

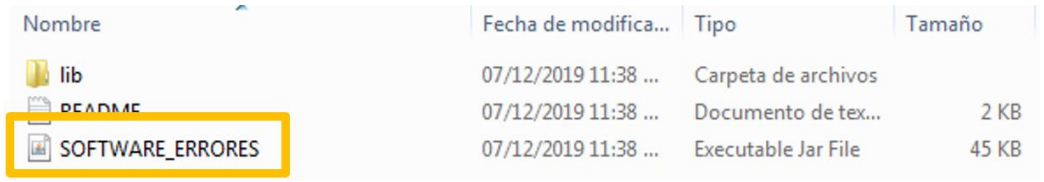




Revisamos la carpeta donde guardamos el proyecto y seleccionamos la opción DIST.

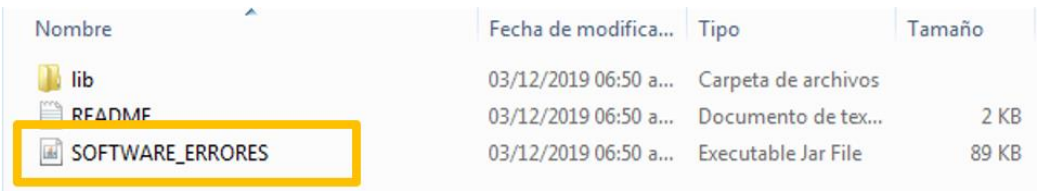


Y observaremos el archivo jar, el cual nos ayudará a ejecutar el programa de manera directa sin la necesidad del abrir el programa Java NetBeans.



2.2. EJECUCIÓN DEL SOFTWARE

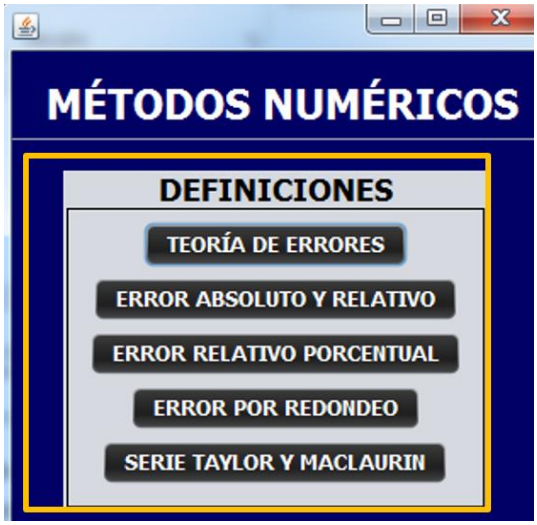
Primero ejecutamos el jar.



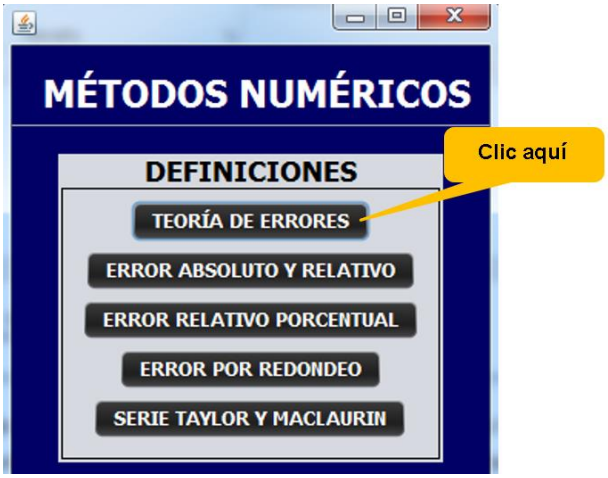
Ejecutamos

Al realizar dicha operación tendremos la siguiente interface, el cual nos muestra todas las dos opciones a utilizar, la primera de definiciones y el segundo que nos ayudará a resolver los ejercicios.

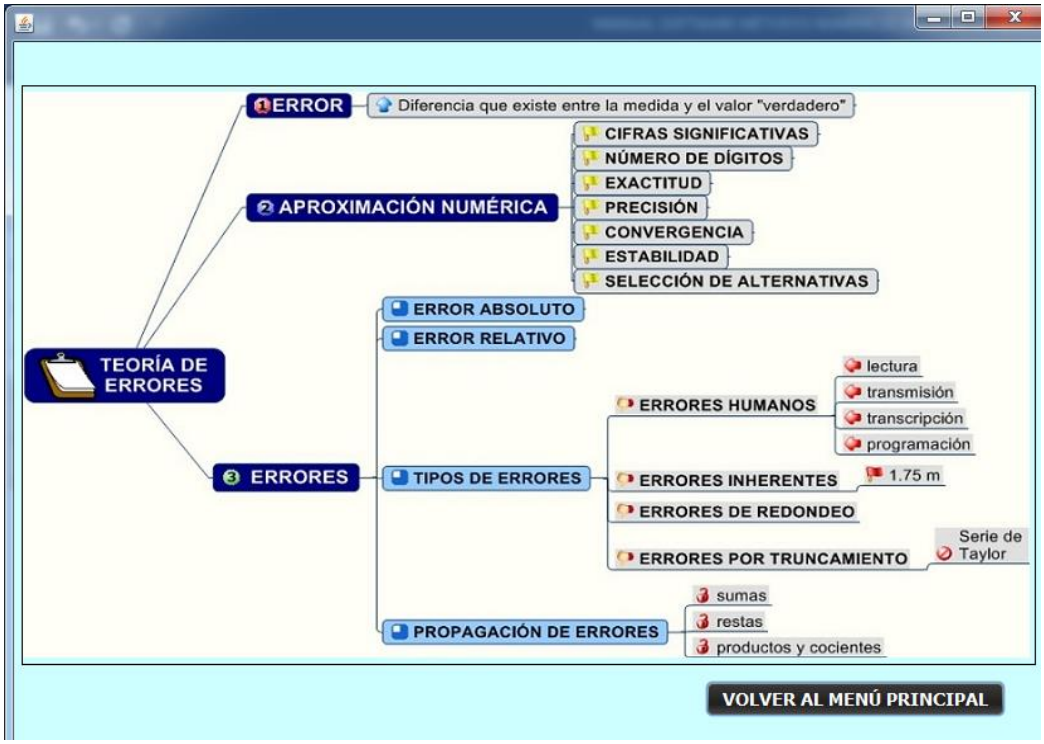
En la primera opción podemos observar las definiciones de los términos más relevantes de la teoría de errores, lo cual nos hará entender las soluciones próximas.



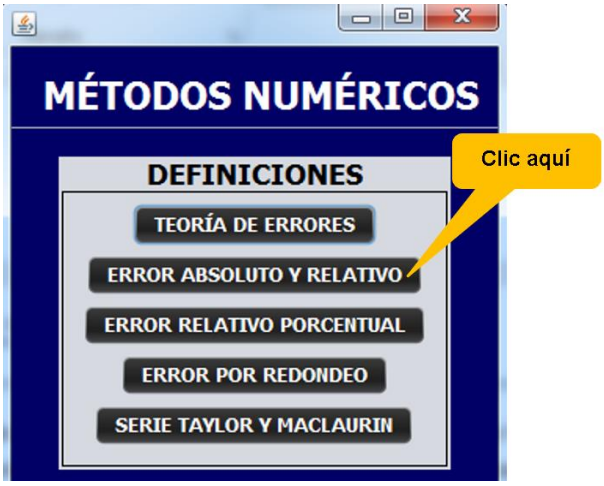
En la primera opción tenemos la definición sobre la teoría de errores, para ello realizamos la siguiente operación:



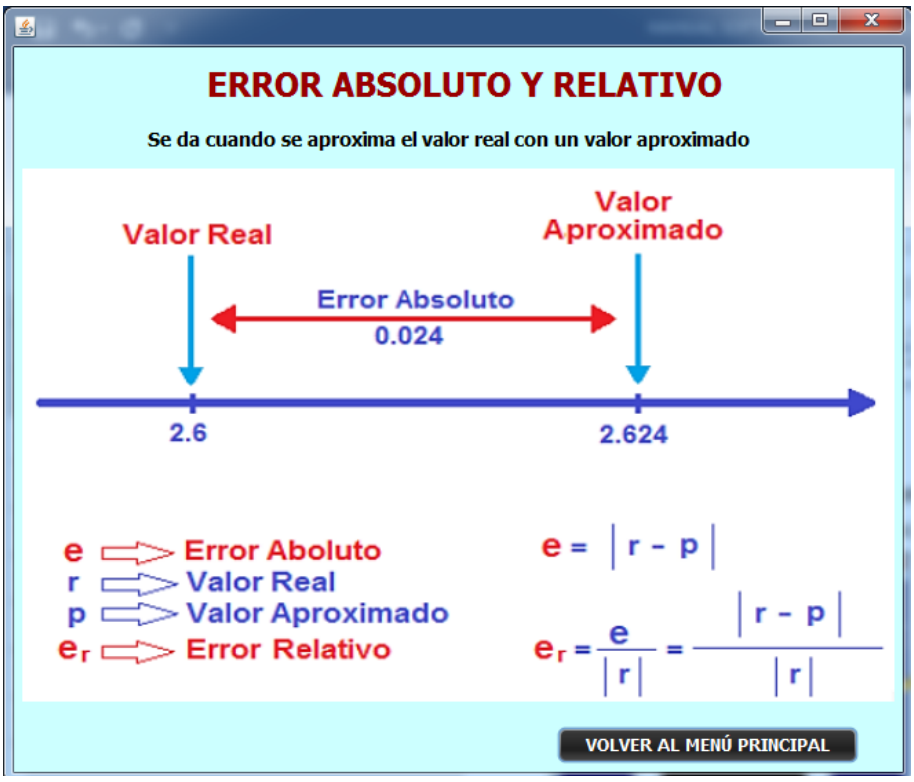
Y observaremos la siguiente interfaz. Damos clic en el botón volver a menú principal para regresar a la interfaz inicial.



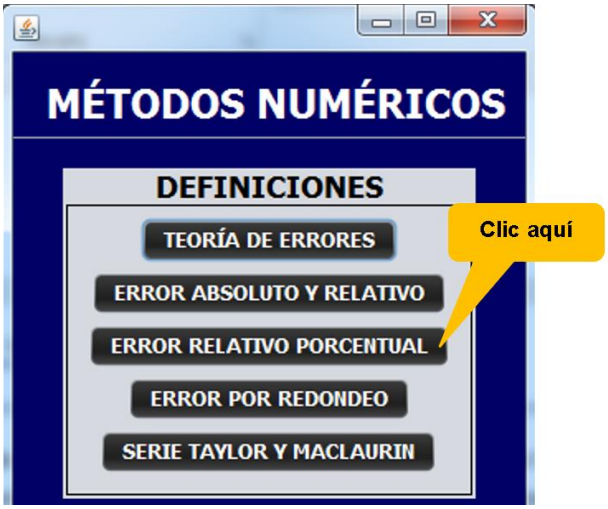
En la segunda opción, encontramos la siguiente definición



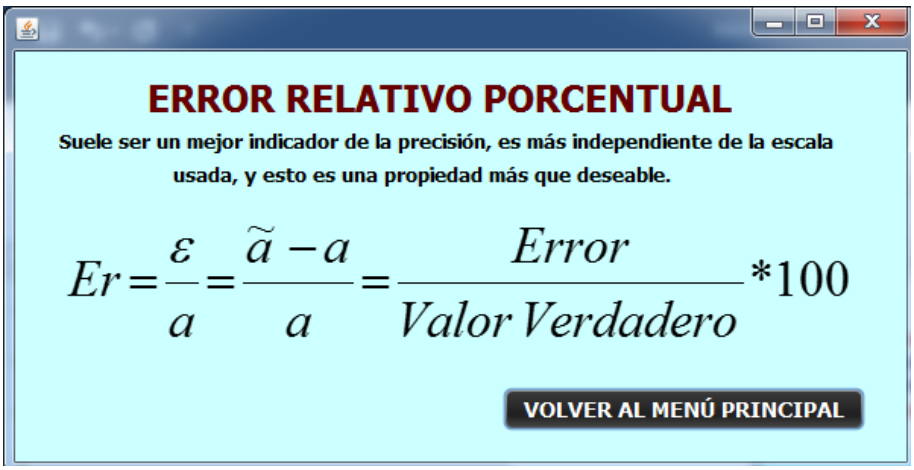
De la misma manera presionamos el botón volver a menú principal para regresar a la interfaz inicial.



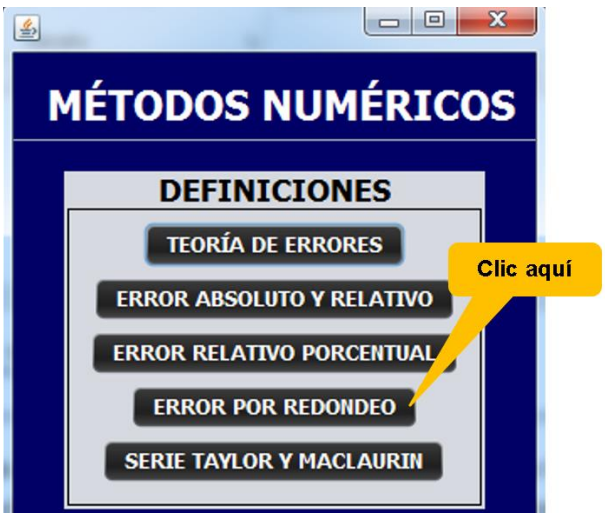
En la tercera opción encontramos la siguiente definición:



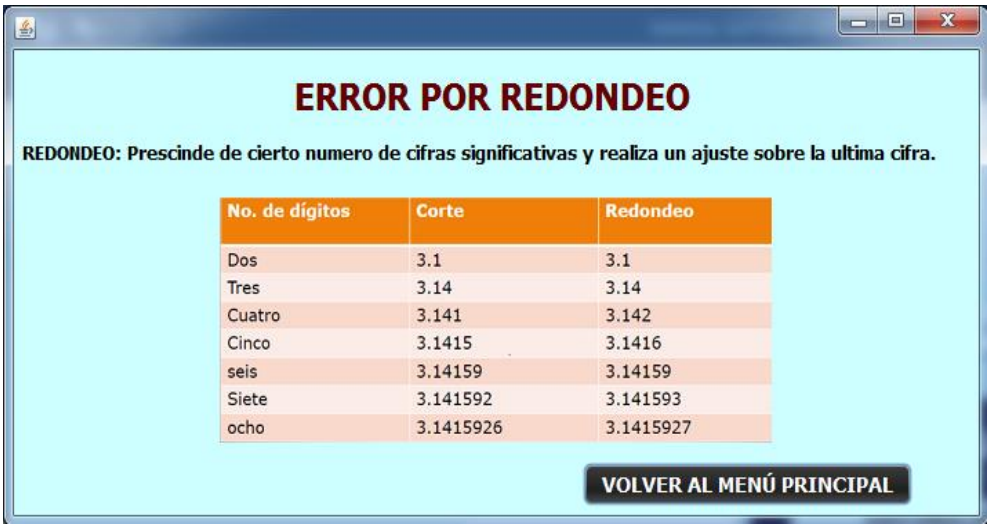
Del mismo modo para regresar a la interfaz inicial se presiona el botón volver a menú principal.



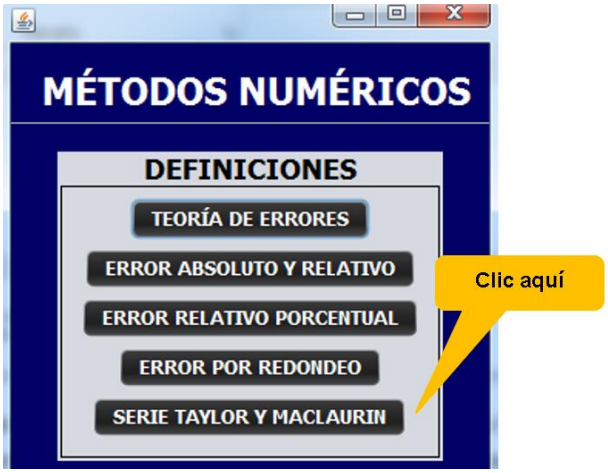
En la cuarta opción encontramos la siguiente definición:



De igual manera presionamos el botón volver al menú principal para regresar a la interfaz inicial.

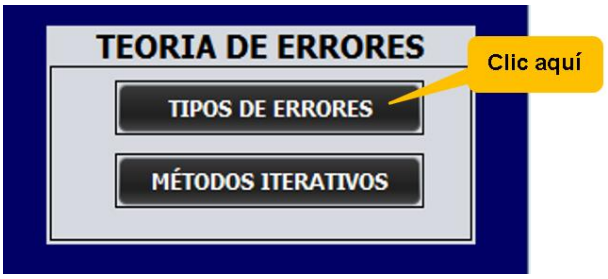


En la última opción encontramos la siguiente definición:



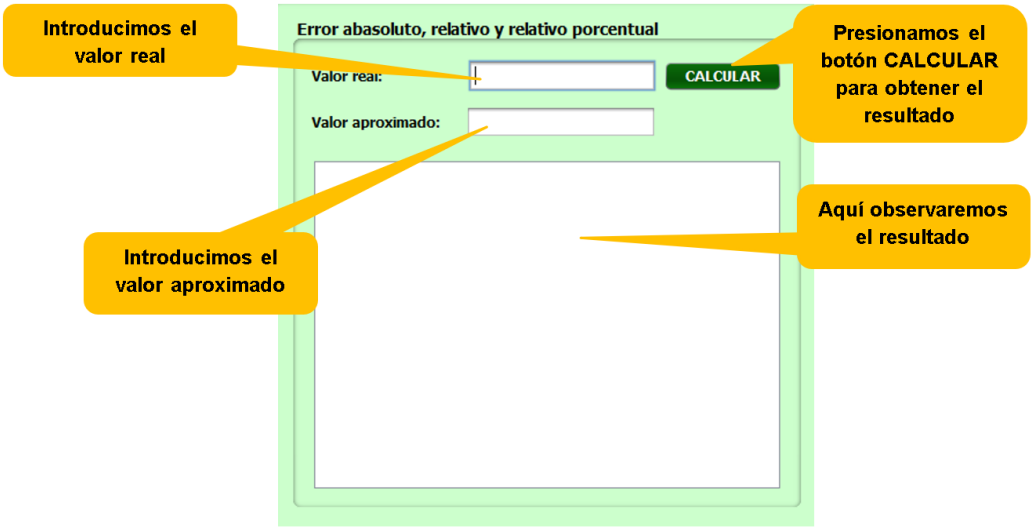
Y de igual manera presionamos el botón volver a menú principal para regresar a la interfaz inicial.

En la segunda parte de la interfaz podemos observar las diferentes opciones que tenemos a utilizar. Damos clic a la primera opción (Tipos de errores).

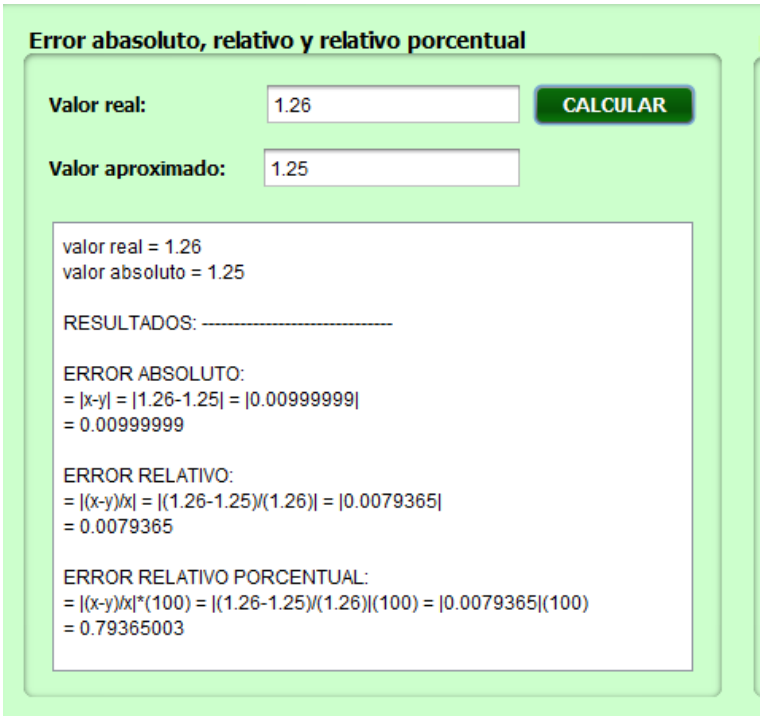


Observamos las dos opciones, la primera parte tenemos: ERROR ABSOLUTO, ERROR RELATIVO, ERROR RELATIVO PORCENTUAL.

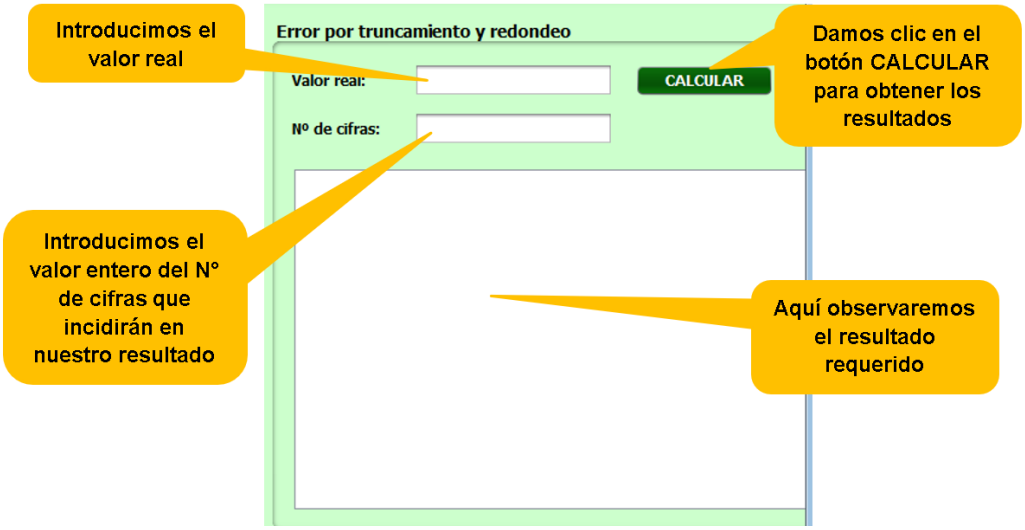
Entonces introducimos los datos y presionamos el botón CALCULAR para obtener dichos resultados.



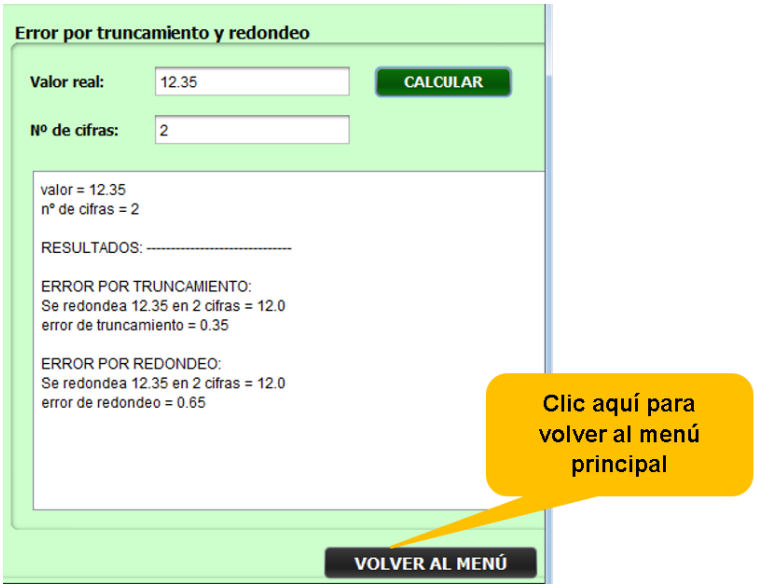
Tal como observamos a continuación



En la segunda parte de la interfaz, observamos el error por redondeo o truncamiento, donde también introduciremos los datos correspondientes para obtener los resultados requeridos.

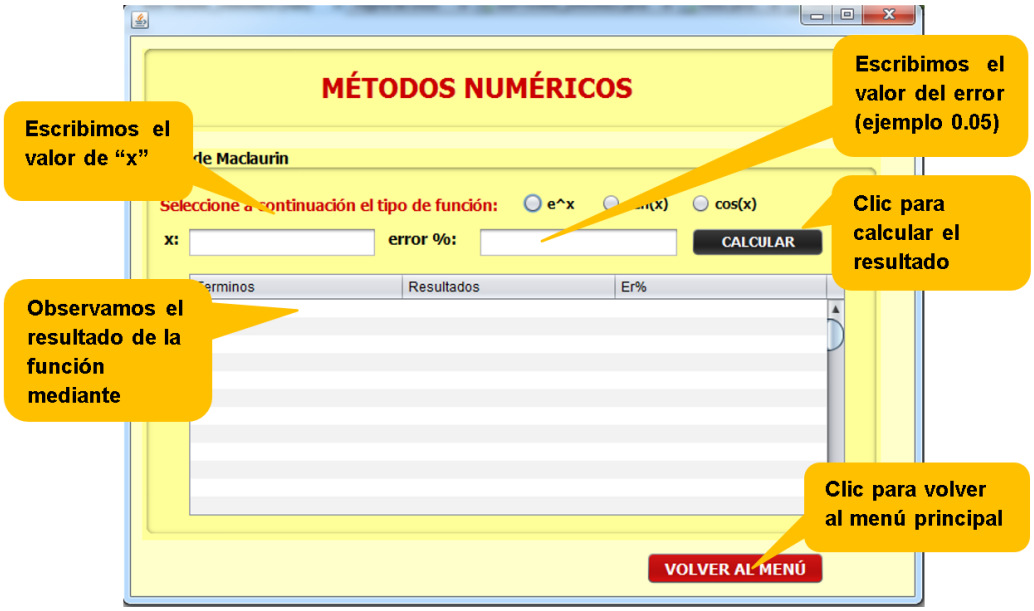


Tal como observamos a continuación:

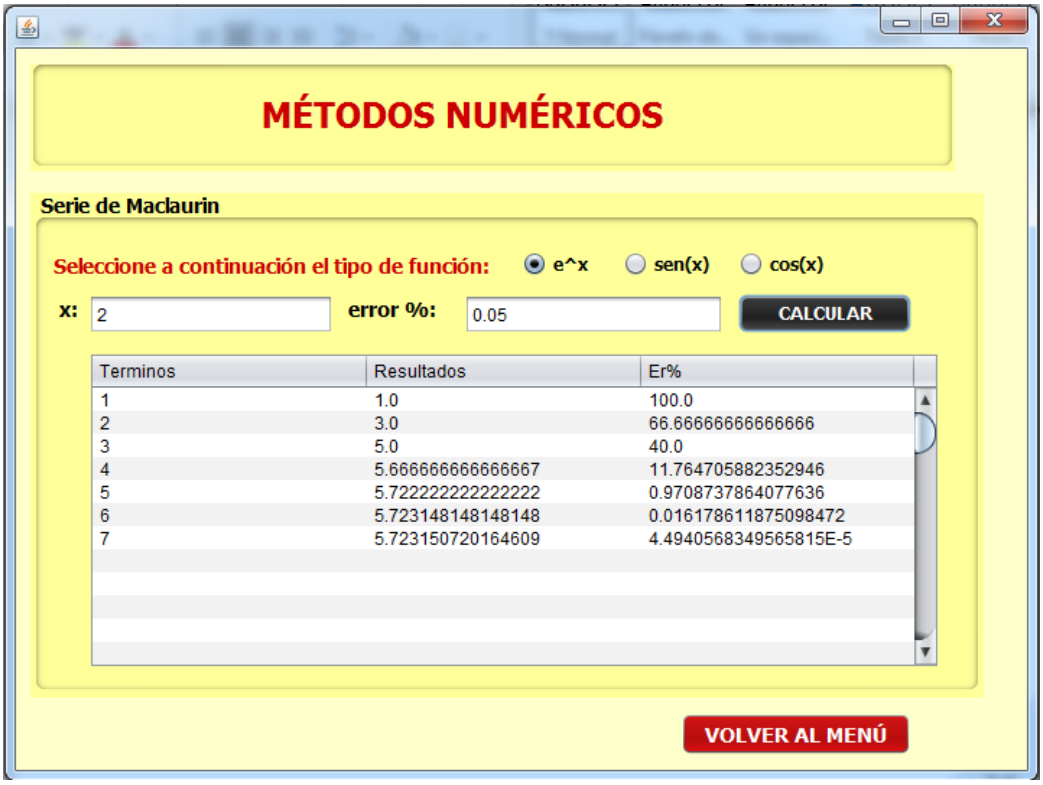


En la segunda opción de la ventana principal tenemos la opción de métodos iterativos (Método Maclaurin)

Al hacer clic en el botón MÉTODOS ITERATIVOS observaremos la siguiente interfaz, donde podremos calcular el resultado de la función mediante el método Maclaurin



Tal como observamos a continuación



Y es así como llegamos al final del manual, donde pudimos observar la creación y codificación del proyecto sobre TEORÍA DE ERRORES en el programa Java NetBeans, el cual nos ayudará a resolver múltiples ejercicios sobre tipos de errores.

SOLUCIÓN NUMÉRICA DE ECUACIONES

INTEGRANTES:

ORTIZ CRUZ, VICTOR ALBERTO

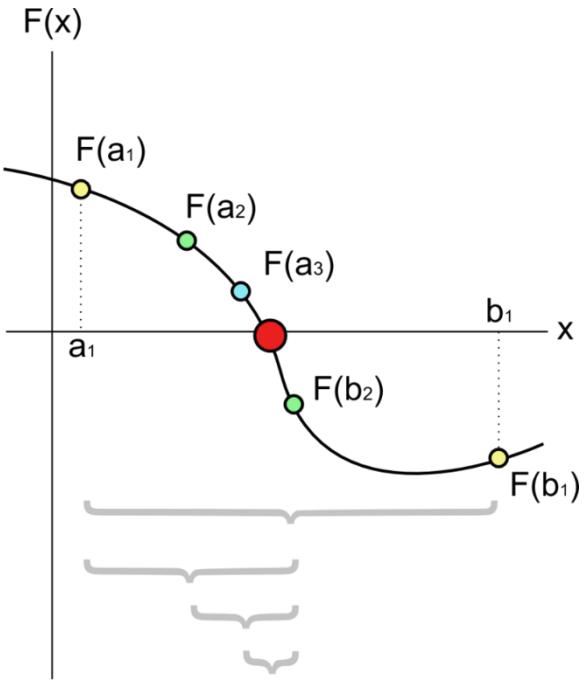
VALDIVIA ANDRES, EDWARD

GOMEZ CASTILLO, NEISSER MIGUEL

BREVE TEORIA DE LOS MÉTODOS

1. MÉTODO DE BISECCIÓN

El método de bisección **es un algoritmo de búsqueda de raíces** que trabaja dividiendo el intervalo a la mitad y seleccionando el subintervalo que tiene la raíz.



- Debe existir seguridad sobre la continuidad de la función *f*(*x*) en el intervalo [*a,b*]
- A continuación, se verifica que f(a)*f(b)<0
- Se calcula el punto medio *m* del intervalo [*a,b*] y se evalúa *f*(*m*) si ese valor es igual a cero, ya hemos encontrado la raíz buscada.
- En caso de que no lo sea, verificamos si *f*(*m*) tiene signo opuesto con *f*(*a*) o con *f*(*b*).Se redefine el intervalo [*a, b*] como [*a, m*] ó [*m, b*] según se haya determinado en cuál de estos intervalos ocurre un cambio de signo.

Con este nuevo intervalo se continúa sucesivamente encerrando la solución en un intervalo cada vez más pequeño, hasta alcanzar la precisión deseada.

2. ALGORITMO

Para aplicar el método consideremos tres sucesiones $a_n \leq r_n \leq b_n$ definidas por las siguientes relaciones:

$$r_n = \frac{a_n + b_n}{2}, \quad a_{n+1} = \begin{cases} a_n & \text{si } f(a_n) \cdot f(r_n) < 0 \\ r_n & \text{si } f(a_n) \cdot f(r_n) > 0 \end{cases}, \quad b_{n+1} = \begin{cases} b_n & \text{si } f(b_n) \cdot f(r_n) < 0 \\ r_n & \text{si } f(b_n) \cdot f(r_n) > 0 \end{cases}$$

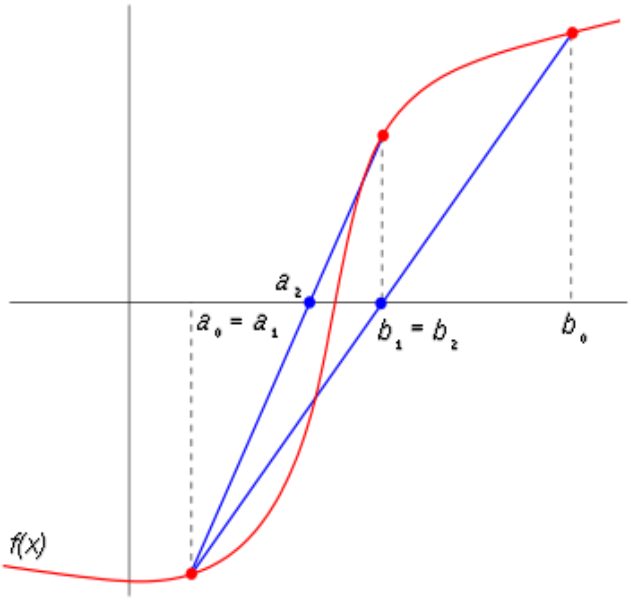
Donde los valores iniciales vienen dados por:

$$a_0 := a, \quad b_0 := b$$

Se puede probar que las tres sucesiones convergen al valor de la única raíz del intervalo:

$$\lim_{n\to\infty} a_n = \lim_{n\to\infty} r_n = \lim_{n\to\infty} b_n$$

3. FALSA POSICIÓN



En cálculo numérico, el método de la regula falsi (regla del falso) o falsa posición **es un método iterativo de resolución numérica de ecuaciones no lineales**. El método combina el método de bisección y el método de la secante.

Como en el método de bisección, se parte de un intervalo inicial $[a_0,b_0]$ con $f(a_0)$ y $f(b_0)$ de signos opuestos, lo que garantiza que en su interior hay al menos una raíz

El algoritmo va obteniendo sucesivamente en cada paso un intervalo más pequeño $[a_k, b_k]$ que sigue incluyendo una raíz de la función f.

A partir de un intervalo $[a_k, b_k]$ se calcula un punto interior c_k:

c_k se determinará con las siguientes formulas:

$$c_k = \frac{\frac{1}{2}f(b_k)a_k - f(a_k)b_k}{\frac{1}{2}f(b_k) - f(a_k)}$$

o

$$c_k = \frac{f(b_k)a_k - \frac{1}{2}f(a_k)b_k}{f(b_k) - \frac{1}{2}f(a_k)}$$

4. MÉTODO DE PUNTO FIJO

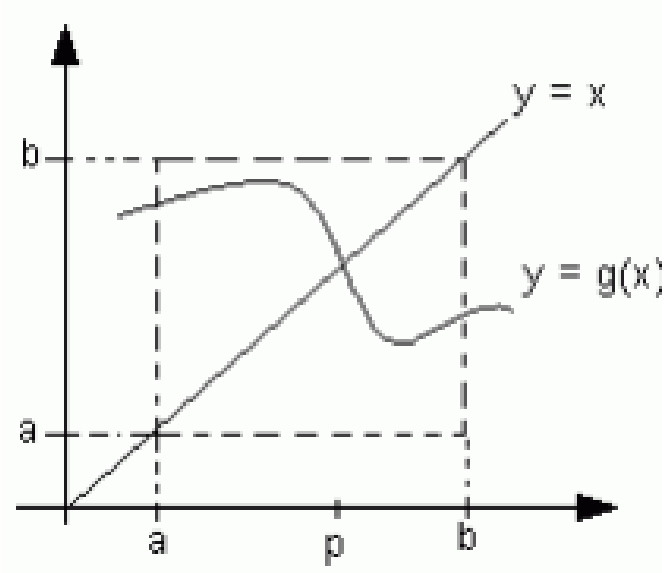
Para hallar raíces de una ecuación $f(x) = 0$, en ciertos casos puede expresarse la ecuación en la forma: $g(x) = x$. A una solución de esta ecuación se la llama punto fijo de la función $g(x)$. Entonces hallar las raíces de la ecuación $f(x) = 0$, equivale a hallar los puntos fijos de $g(x)$. Por eso es importante conocer cuando una función tiene punto fijo (o más de uno) y cómo calcularlo.

Teorema 1.

{1} Si $g\ C[a,b]$ y $g(x) \in [a,b]$ para todo $x \in [a,b]$, entonces g tiene un punto fijo en $[a,b]$. Este punto fijo no tiene por qué ser único.

{2} Pero si además, $g'(x)$ existe en (a,b) y $|g'(x)| \leq k < 1$ para todo $x \in (a,b)$.

Entonces g tiene un único punto fijo $[a,b]$.



Ejemplo 1:

Sea $g(x) = \frac{x^2-1}{3}$ en el intervalo $[-1, +1]$. Esta función es continua y diferenciable en este intervalo, y toma valores iguales en sus extremos, $g(-1) = g(1) = 0$, por el teorema del valor extremo existe un minimo absoluto en $[a, b]$, que se encuentra concretamente en $x = 0,\ g(0) = \frac{-1}{3}$. Por otra parte, el máximo absoluto de g se encuentra en los extremos.

Por tanto, la funcion $g(x)$ cumple la condicion $\{1\}$ → hay algun punto fijo.

Además existe derivada de $g(x)$ en $(-1, +1)$:

$$|g'(x)| = \left|\frac{2}{3}x\right| \leq \frac{2}{3} < 1$$

O sea, que $g(x)$ cumple tambien la propiedad $\{2\}$, lo que significa que hay un unico punto fijo en el intervalo $[-1, +1]$.

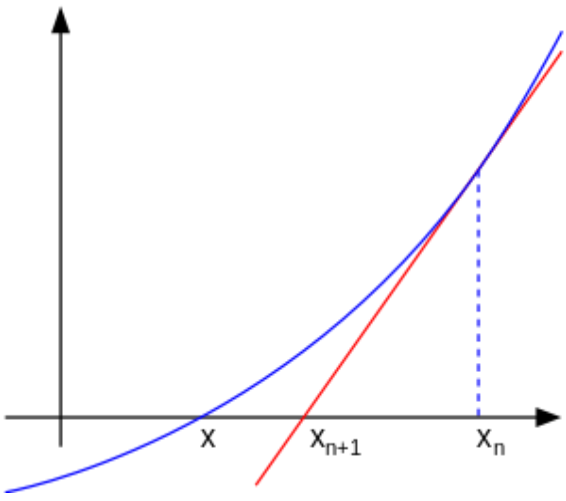
Para este caso concreto puede hallarse el "punto fijo", sin más que resolver la ecuación:

$$p = g(p) \rightarrow p = \frac{p^2 - 1}{3}$$

La solución exacta es $p = \frac{3-\sqrt{13}}{2}$.

5. MÉTODO DE NEWTON-RAPHSON

Es un algoritmo para encontrar aproximaciones de los ceros o raíces de una función real. También puede ser usado para encontrar el máximo o mínimo de una función, encontrando los ceros de su primera derivada.



Ejemplo:

Consideremos el problema de encontrar un número positivo x tal que $cos(x) = x^3$. Podríamos tratar de encontrar el cero de $f(x) = cos(x) - x^3$.

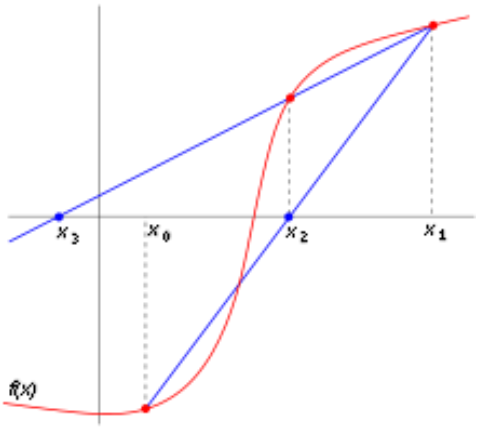
Sabemos que $f'(x) = -sin(x) - 3x^2$. Ya que $cos(x) \leq 1$ para todo x y $x^3 > 1$ para $x > 1$, deducimos que nuestro cero está entre 0 y 1. Comenzaremos probando con el valor inicial $x_0 = 0{,}5$.

$$
\begin{aligned}
x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} = 0,5 - \frac{\cos(0,5) - 0,5^3}{-\sin(0,5) - 3\times 0,5^2} &&= 1,112141637097 \\
x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} &&\vdots \quad = 0,909672693736 \\
x_3 &\quad \vdots &&\vdots \quad = 0,867263818209 \\
x_4 &\quad \vdots &&\vdots \quad = 0,865477135298 \\
x_5 &\quad \vdots &&\vdots \quad = 0,865474033111 \\
x_6 &\quad \vdots &&\vdots \quad = 0,865474033102
\end{aligned}
$$

Los dígitos correctos están subrayados. En particular, x_6 es correcto para el número de decimales pedidos. Podemos ver que el número de dígitos correctos después de la coma se incrementa desde 2 (para x_3) a 5 y 10, ilustrando la convergencia cuadrática.

6. MÉTODO DE LA SECANTE

El método de la secante **es un algoritmo de la raíz de investigación que utiliza una serie de raíces de las líneas secantes para aproximar mejor la raíz de una función.**



El método se define por la *relación de recurrencia*:

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

Como se puede ver, este método necesitará dos aproximaciones iniciales de la raíz para poder inducir una pendiente inicial.

EJERCICIO DE EJEMPLO

Utilice el método de la secante para encontrar una raíz real de la ecuación polinomial:

$$f(x) = x^3 + 2x^2 + 10x - 20 = 0.$$

Utilizando la ecuación:

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

Obtenemos:

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})(x_n^3 + 2x_n^2 + 10x_n - 20)}{(x_n^3 + 2x_n^2 + 10x_n - 20) - ((x_{n-1}^3 + 2x_{n-1}^2 + 10x_{n-1} - 20))}$$

$$x_2 = 1 - \frac{(1-0)(1^3+2(1)^2+10(1)-20)}{(1^3+2(1)^2+10(1)-20)-(0^3+2(0)^2+10(0)-20)} = 1.53846$$

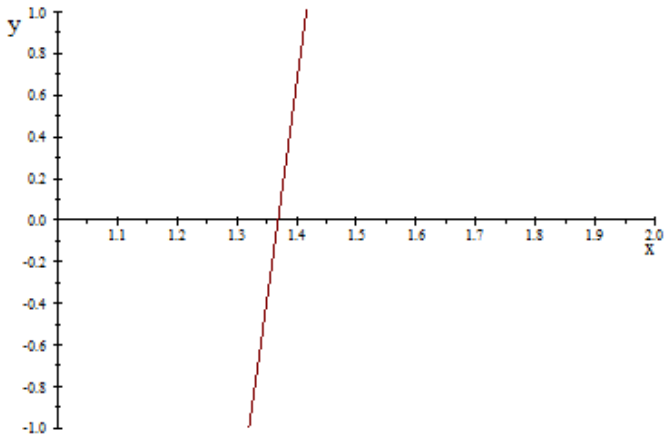
Los valores posteriores son los siguientes:

n	x_n	$\lvert x_{n+1} - x_n \rvert$
0	0.00000	
1	1.00000	1.00000
2	1.53846	0.53846
3	1.35031	0.18815
4	1.36792	0.01761
5	**1.36881**	**0.00090**

Ahí tenemos el resultado, cuando:

$$|x_{n-1} - x_n| \leq \epsilon = 10^{-3}$$

Comprobando el resultado graficando la función utilizando software obtenemos:



7. MÉTODO DE MULLER

Este método **utilizado para encontrar raíces de ecuaciones con raíces múltiples**, y consiste en obtener los coeficientes de la parábola que pasa por tres puntos elegidos. Dichos coeficientes son sustituidos en la formula cuadrática para obtener el valor donde la parábola

intersecta al eje X; es decir, la raíz estimada. La aproximación se puede facilitar, si se escribe la ecuación de la parábola en una forma conveniente.

> **Una de las mayores ventajas de este método, es que al trabajar con la formula cuadrática es posible localizar tanto raíces reales, como raíces complejas.**

Fórmula

Los tres valores iniciales necesitados son denotados como $x_k,\ x_k - 1$ y $x_k - 2$. La parábola pasa a través de los puntos: $(x_k,\ f(x_k)), (x_k - 1,\ f(x_k - 1))$ y $(x_k - 2,\ f(x_k - 2))$, si se escribe en la forma de *Newton*, entonces:

$$y = f(x_k) + (x - x_k)f[x_k, x_{k-1}] + (x - x_k)(x - x_{k-1})f[x_k, x_{k-1}, x_{k-2}]$$

donde $f[x_k, x_{k-1}]$ y $f[x_k, x_{k-1}, x_{k-2}]$ denotan restas divididas. Esto puede ser escrito como:

$$y = f(x_k) + w(x - x_k) + f[x_k, x_{k-1}, x_{k-2}](x - x_k)^2$$

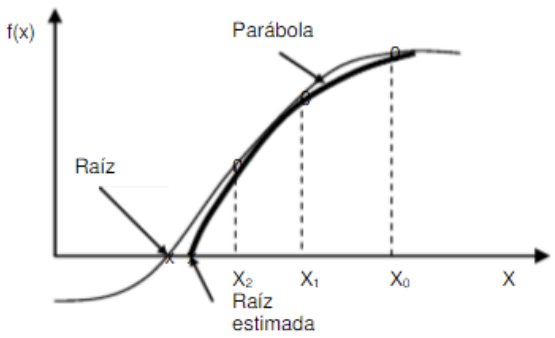
Donde

$$w = f[x_k, x_{k-1}] + f[x_k, x_{k-2}] - f[x_{k-1}, x_{k-2}].$$

La próxima iteración está dada por la raíz que brinda la ecuación $y = 0$.

$$x_{k-1} = x_k - \frac{2f(x_k)}{w \pm \sqrt{w^2 - 4f(x_k)f[x_k, x_{k-1}, x_{k-2}]}}$$

Representación gráfica:



8. MÉTODO DE BAIRSTOW

En análisis numérico, el **método de Bairstow** es un algoritmo eficiente de búsqueda de las raíces de un polinomio real de grado arbitrario.

Es un método iterativo, basado en el método de Müller y de Newton Raphson.

Dado un polinomio $f_n(x)$ se encuentran dos factores, un polinomio cuadrático

$$f_2(x) = x^2 - rx - s \text{ y } f_{n-2}(x)$$

El procedimiento general para el ***método de Bairstow*** es el siguiente. Dado:

$$f_n(x) \; y \; r_0 \; y \; s_0$$

a) Utilizando el método de Newton Raphson se calcula:

$f_2(x) = x^2 - rx - s$ y $f_{n-2}(x)$, tal que, el residuo de: $\frac{f_n(x)}{f_2(x)}$ sea igual a cero.

b) Se determinan las raíces $f_2(x)$, utilizando la fórmula general.

c) Se calcula $f_{n-2}(x) = \frac{f_n(x)}{f_2(x)}$

d) Se hace $f_n(x) = f_{n-2}(x)$

e) Si el grado del polinomio es mayor que tres regresamos al paso 2; en caso contrario, terminamos.

La principal diferencia de este método, respecto a otros, es que permite calcular todas las raíces de un polinomio (reales e imaginarias). Para calcular la división de polinomios, hacemos uso de la división sintética. Así dado:

$$f_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$$

Al dividir entre , se tiene como resultado el siguiente polinomio:

$$f_{n-2}(x) = b_n x^{n-2} + b_{n-1}x^{n-3} + \cdots + b_3 x + b_2$$

con un residuo, , el residuo será cero solo si , lo son. Los términos b, se calculan utilizando división sintética, la cual puede resolverse utilizando la siguiente relación de recurrencia:

$$b_n = a_n,$$

$$b_{n-1} = a_{n-1} + rb_n,$$

$$b_i = a_i + rb_{i+1} + sb_{i+2},$$

Una manera de determinar los valores de r y s que hacen cero el residuo es utilizar el método de Newton-Raphson. Para ello necesitamos una aproximación lineal de b_1 y b_0, respecto a r y s la cual calculamos utilizando la serie de Taylor

$$b_1(r+dr, s+ds) = b_1 + \frac{\partial b_1}{\partial r}dr + \frac{\partial b_1}{\partial s}ds$$

$$b_0(r+dr, s+dr) = b_0 + \frac{\partial b_0}{\partial r}dr + \frac{\partial b_0}{\partial s}ds$$

donde los valores de r y s están dados y se calculan los incrementos dr y ds que hacen a $b_1(r+dr, s+ds)$ y $b_0(r+dr, s+dr)$ igual a cero. El sistema de ecuaciones que se tiene que resolver es:

$$\frac{\partial b_1}{\partial r}dr + \frac{\partial b_1}{\partial s}ds = -b_1$$
$$\frac{\partial b_0}{\partial r}dr + \frac{\partial b_0}{\partial s}ds = -b_0$$

Bairtow muestra que las derivadas parciales pueden obtener haciendo un procedimiento similar a la división sintética, así:

$$c_n = b_n$$

$$c_{n-1} = b_{n-1} + rc_n$$

$$c_i = b_i + rc_{i+1} + sc_{i+2}$$

Donde:

$$\frac{\partial b_0}{\partial r} = c_1$$

$$\frac{\partial b_1}{\partial r} = \frac{\partial b_0}{\partial s} = c_2$$

$$\frac{\partial b_1}{\partial s} = c_3$$

9. MÉTODO DE APROXIMACIÓN GRÁFICA

El método de la Aproximación gráfica es el método más sencillo pero el menos indicado, debido a que solo **vas a ir dando valores a la incógnita de una función hasta llegar al punto más cercano de la raíz.**

Fórmula

Para este Método no hay formula alguna solo se utiliza la ecuación deseada, por ejemplo $f(x) = x^3 + 2x^2 + 10x - 20 = 0$, y se le daría valor a x hasta llegar al resultado deseado.

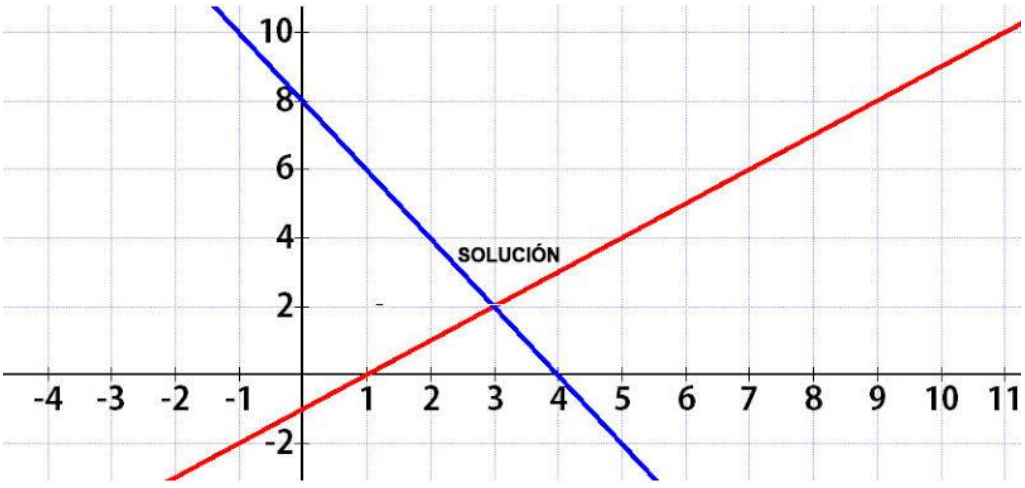
Algoritmo

Aquí se pondrá un ejemplo de cómo llevar acabo la aproximación gráfica:

$$f(x) = x^3 + 2x^2 + 10x - 20 = 0$$

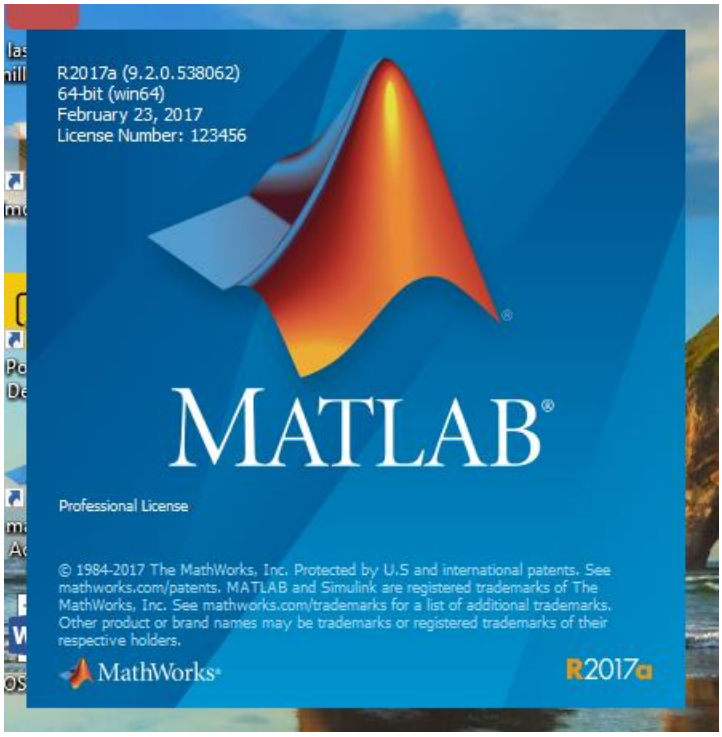
x	$f(x)$
4	68
3	37
2	12
1	-7

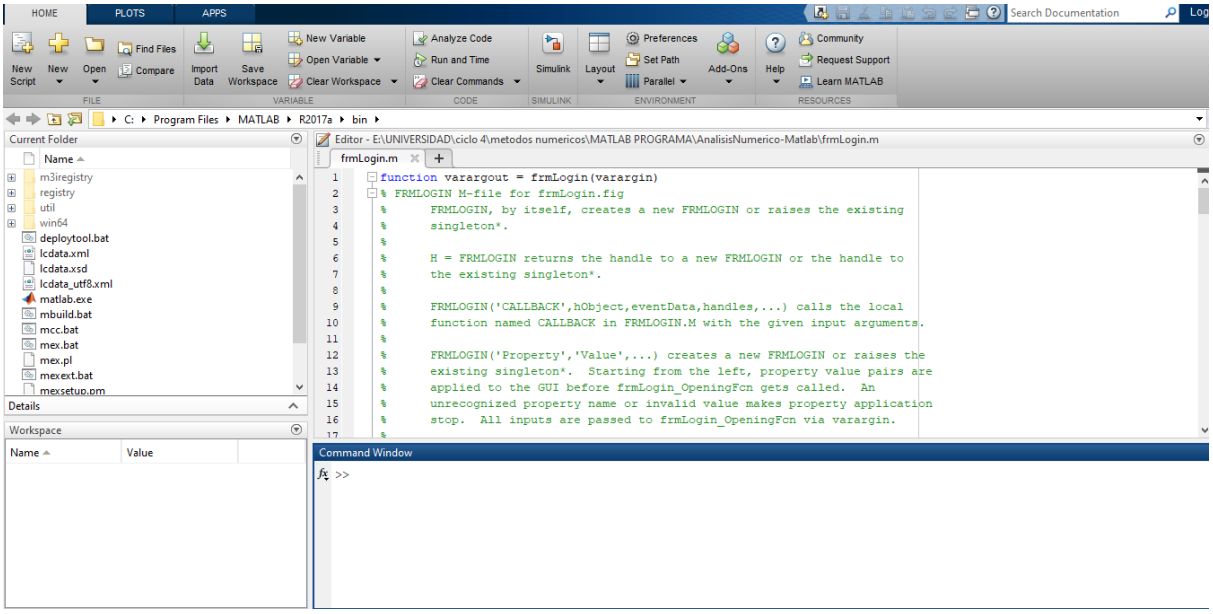
Representación gráfica



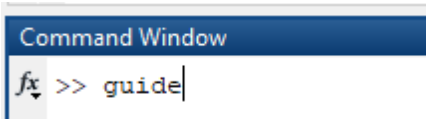
¿COMÓ PROGRAMO EN MATLAB SOLUCIÓN DE ECUACIONES NO LINEALES?

Abrimos el matlab y esperemos a que cargue

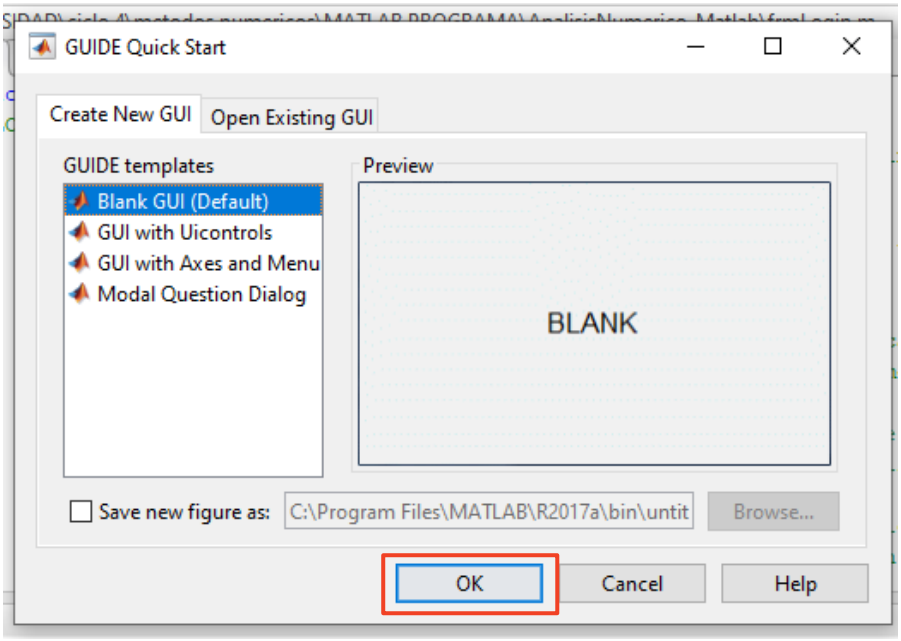




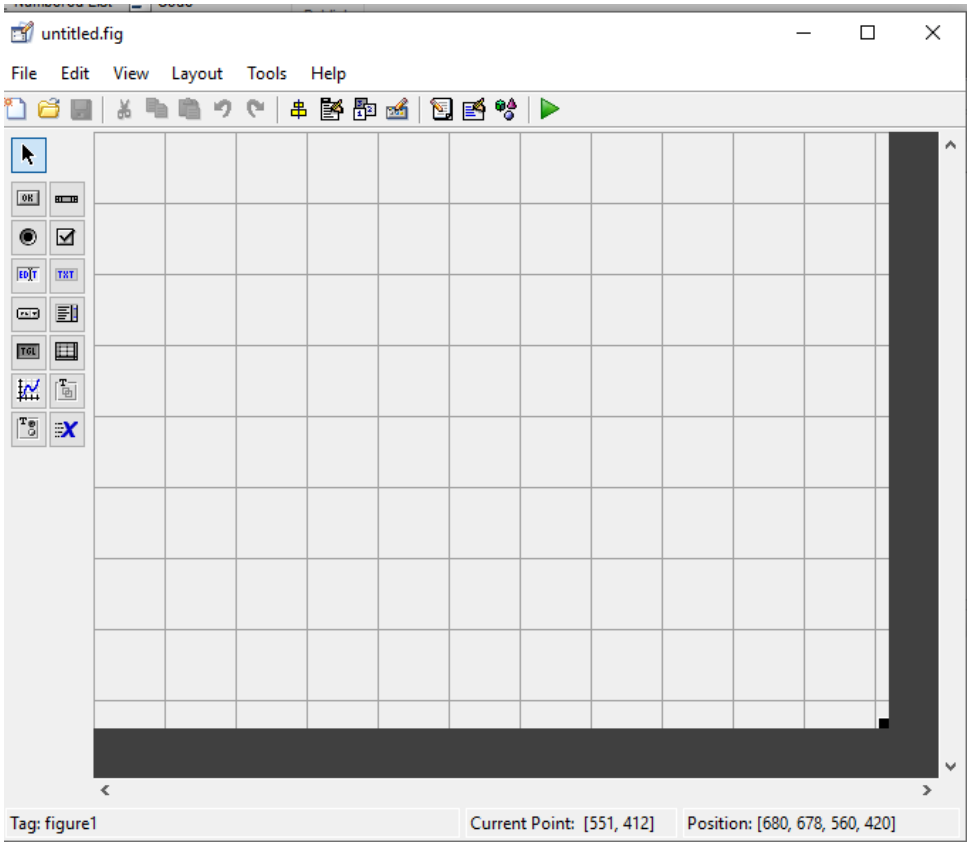
En la parte de commant window escribimos >> *guide* luego esperamos a que cargue.



Se abrira una ventana para nuestro caso utilizaremos la primera selección luego clic en *OK*.

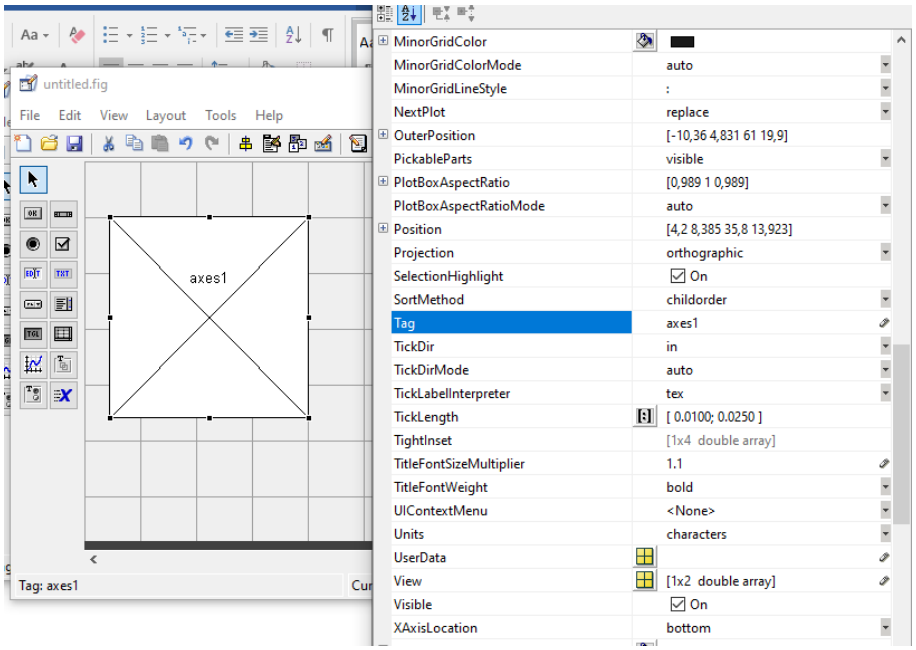


Te mostrara la siguiente ventana:

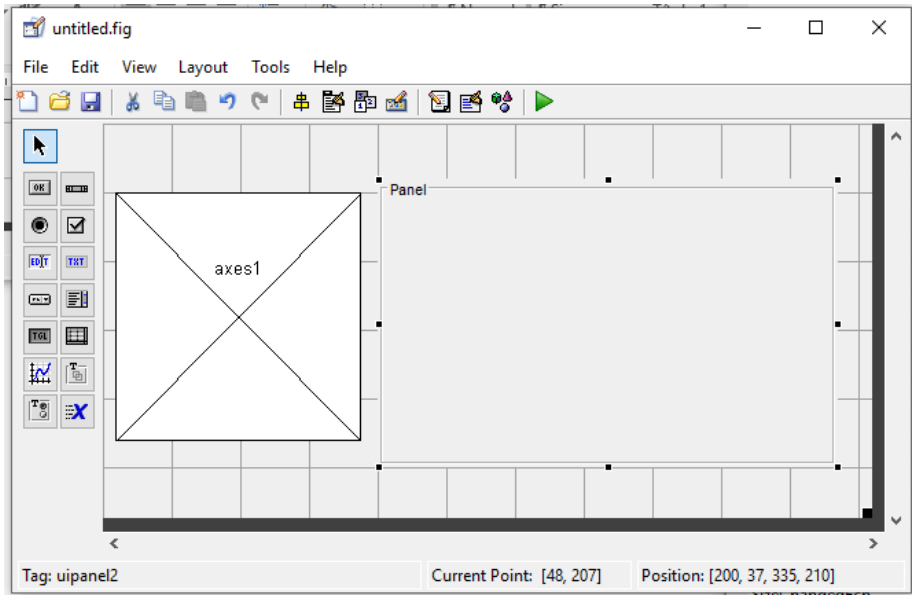


¿CÓMO CREAR INICIO DE SESIÓN?

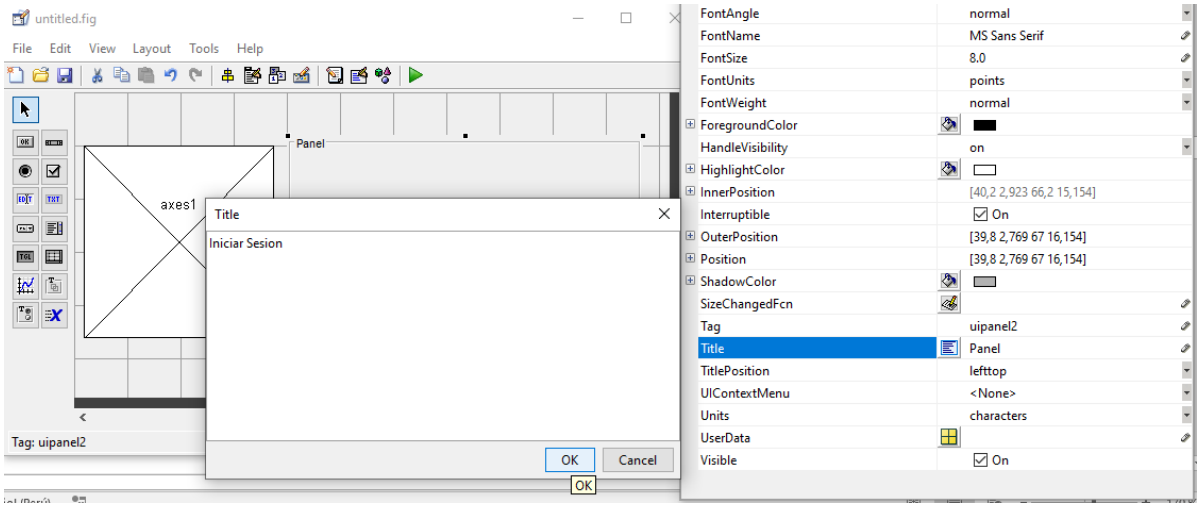
Creamos el ***axes 1***



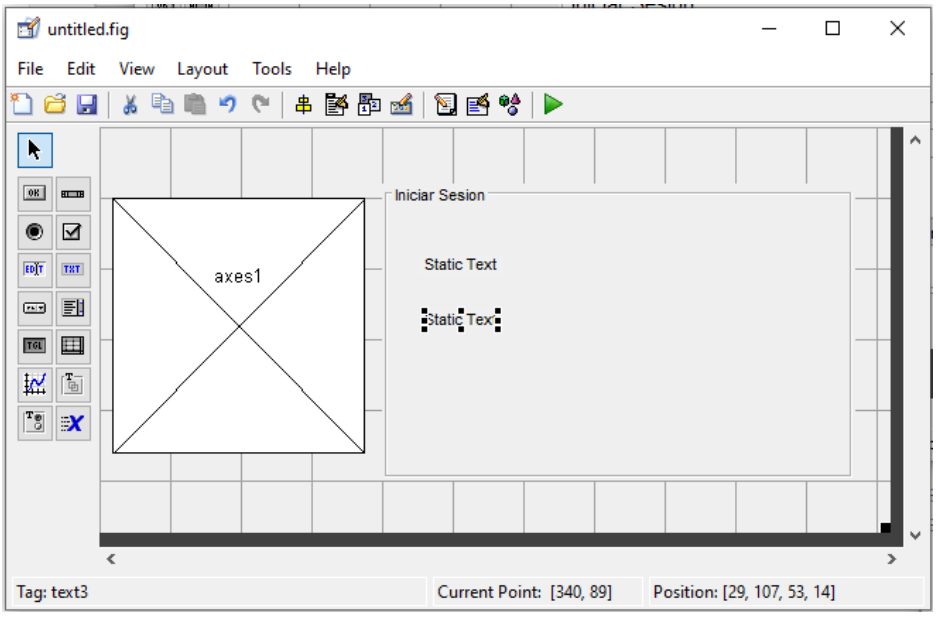
Creamos un panel



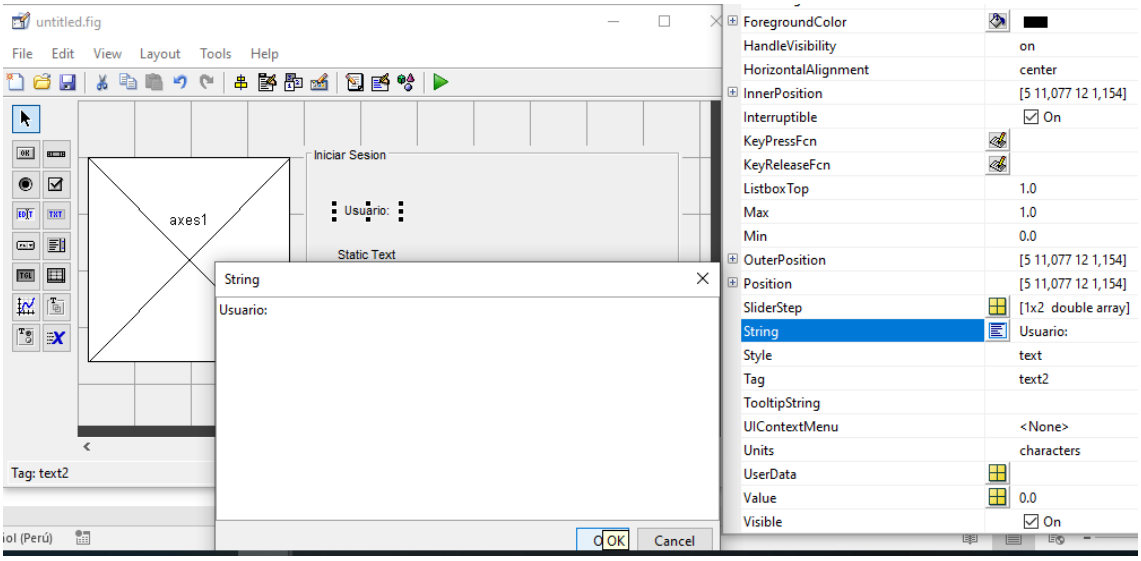
En el ***title*** digitamos ***Iniciar Sesión***, clic en ***OK***.



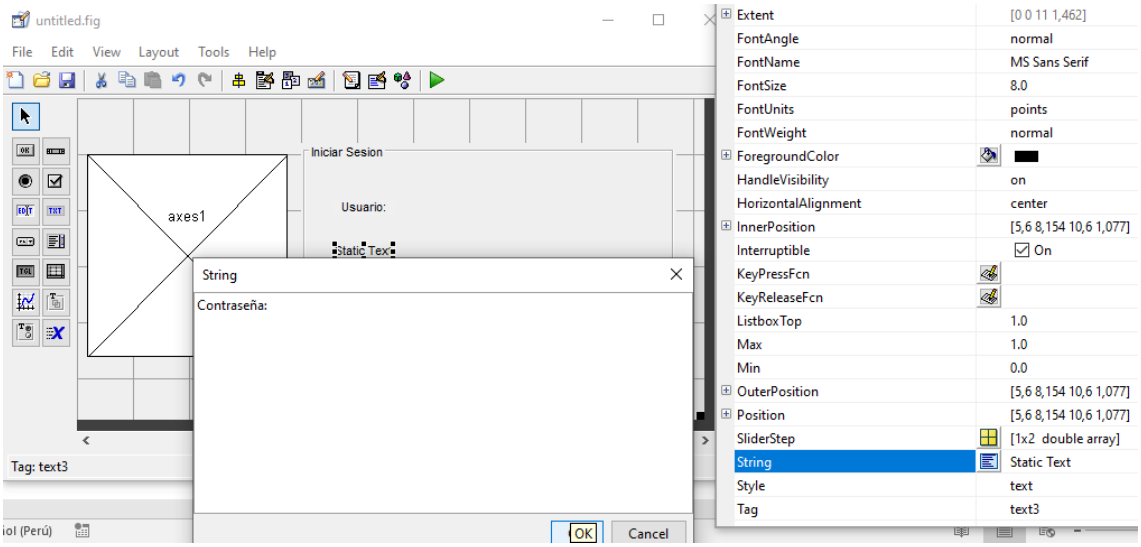
Luego creamos ***2 Static text.***



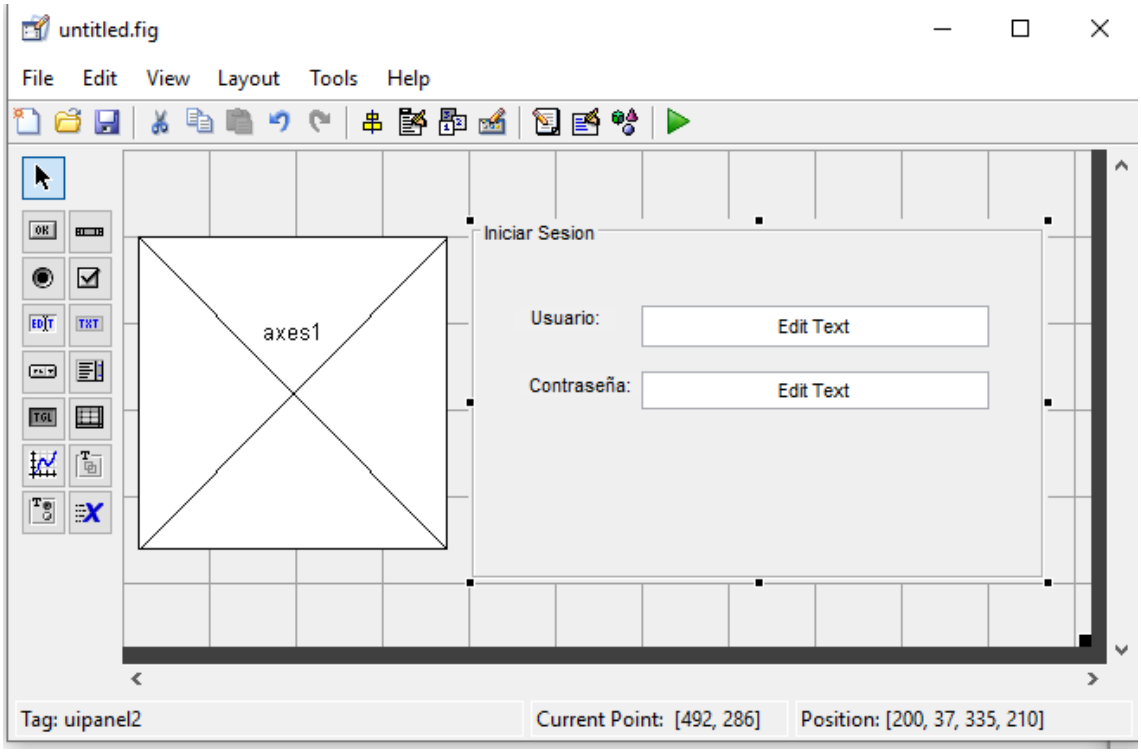
En ***String*** digitamos ***Usuario:*** luego clic en ***OK***.



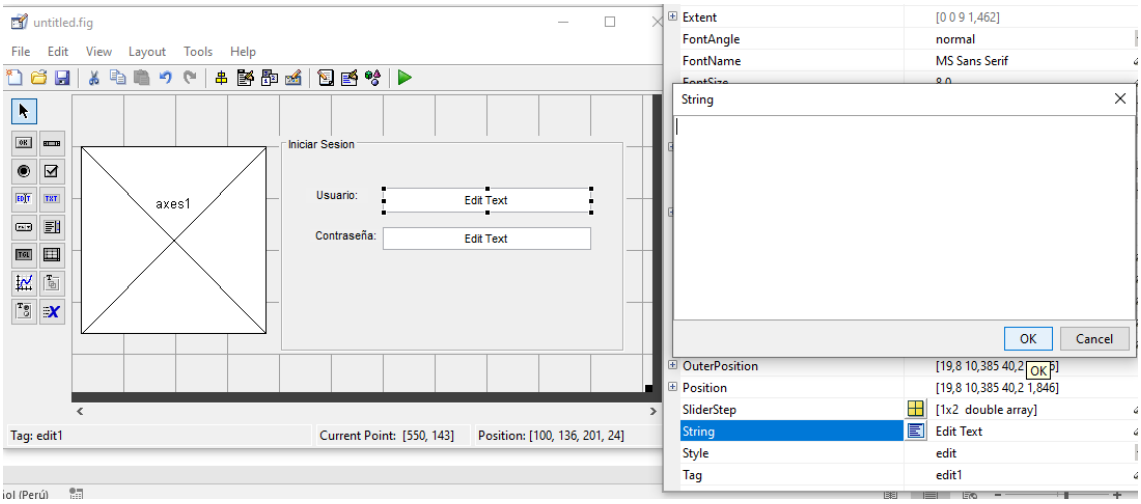
Lo mismo para el siguiente solo que en la parte de ***String*** digitamos ***Contraseña:*** luego clic en ***OK***.



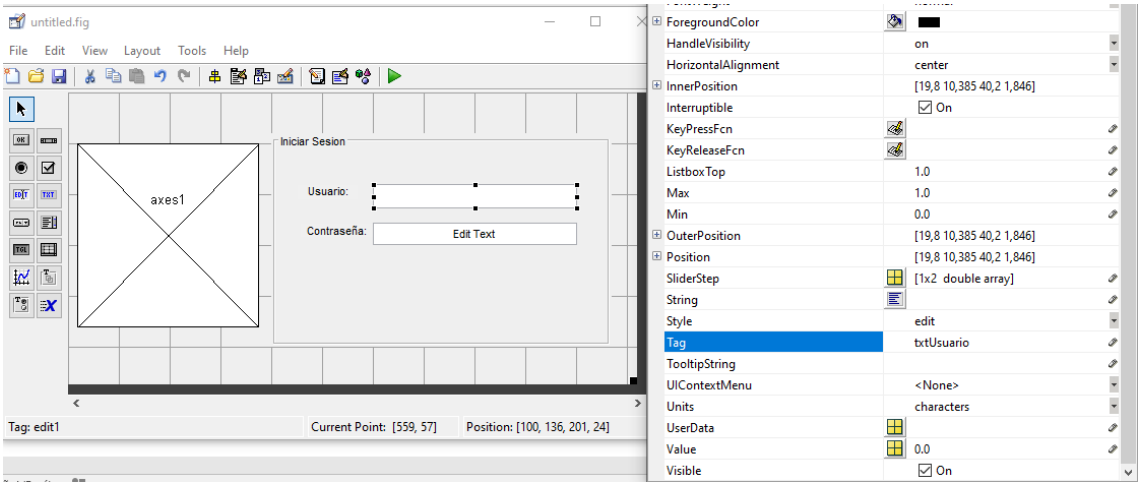
Creamos ***2 Edit Text***



En la parte de ***String*** eliminamos todo luego clic en ***OK***.



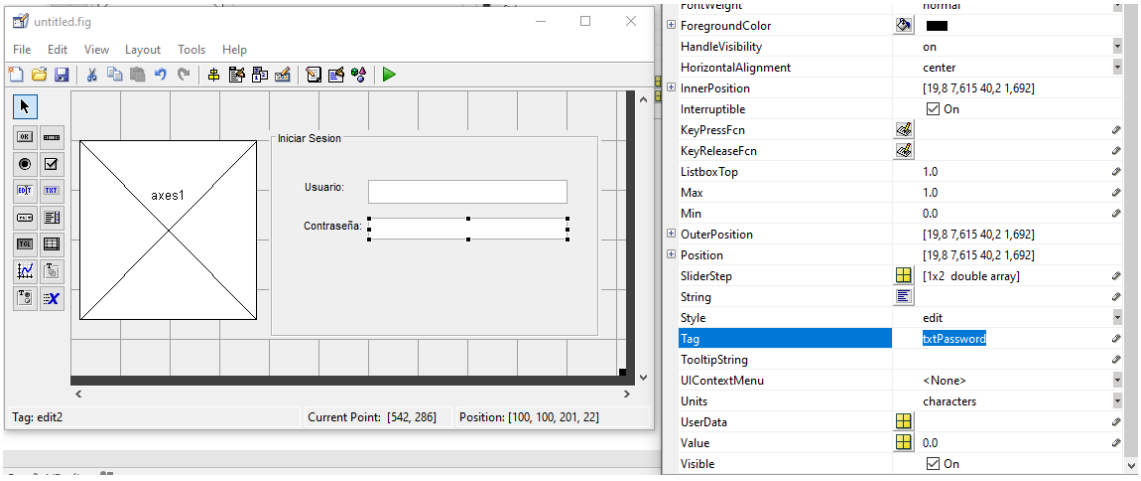
Luego en ***tag*** digitamos ***txtUsuario.***



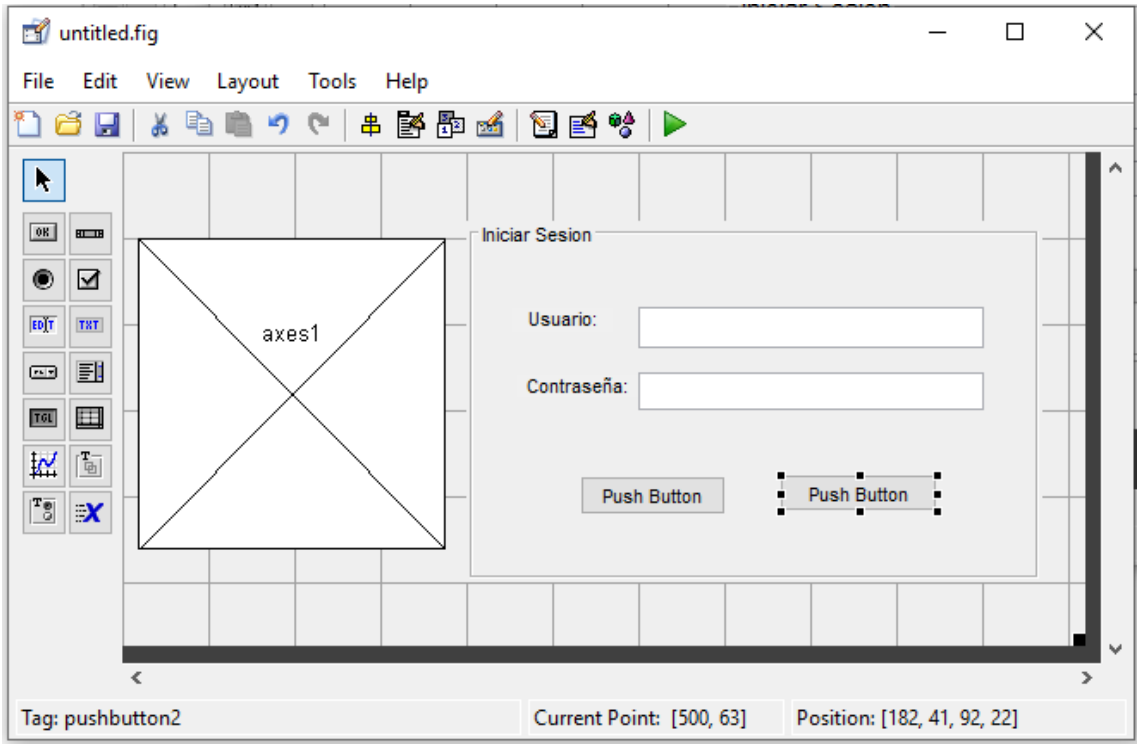
Lo mismo para el siguiente ***Edit Text*** hacemos lo siguiente

En ***String*** eliminamos todo, clic en ***OK***

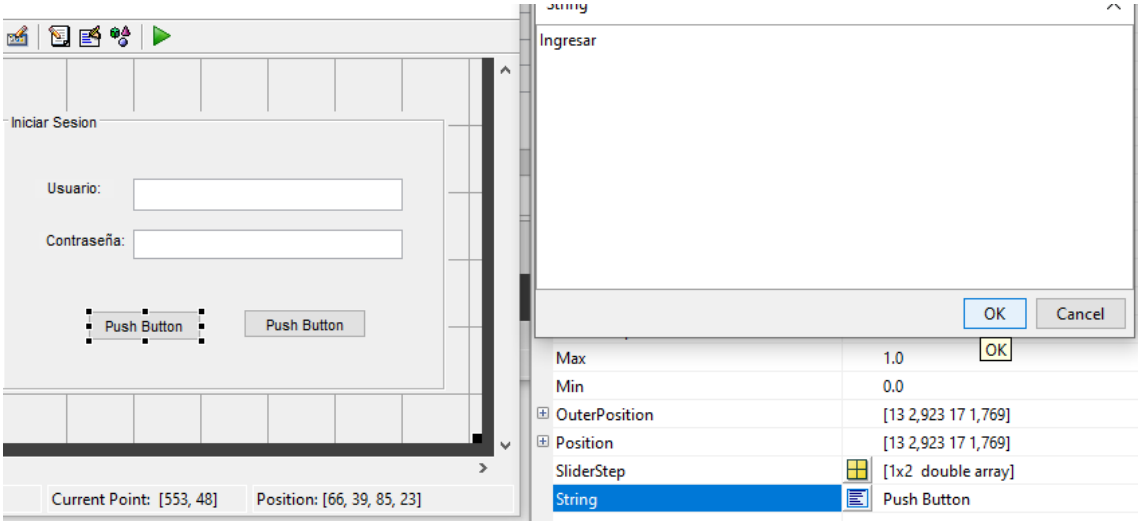
En ***Tag*** digitamos ***txtPasword.***



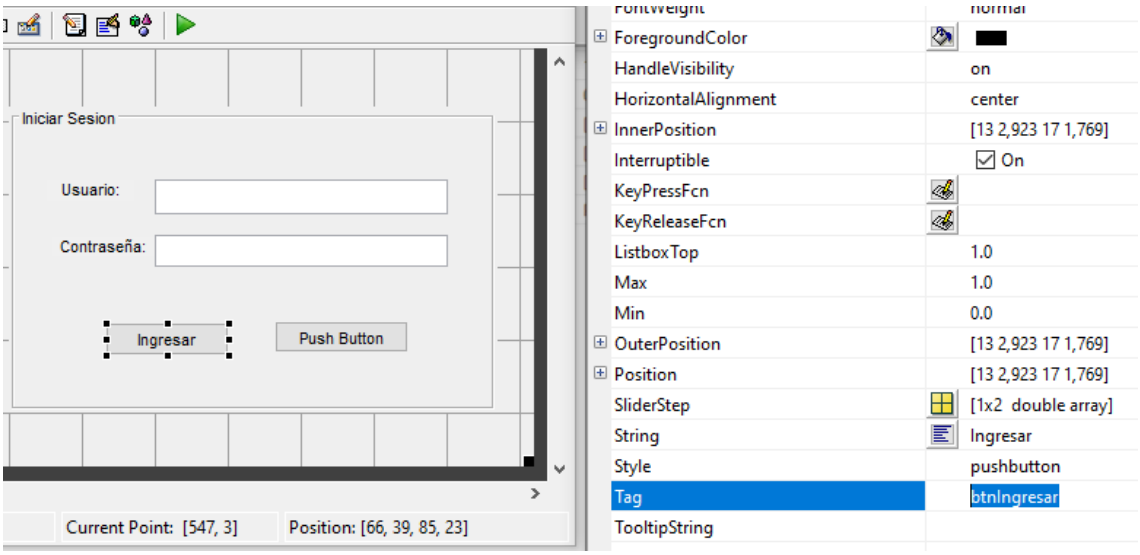
Creamos ***2 Push Buttom***



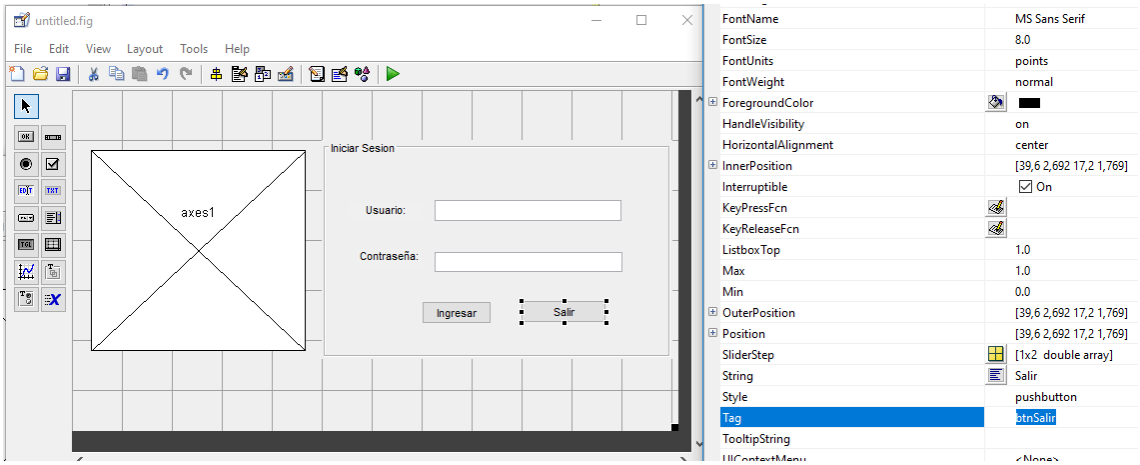
Para el primero digitamos en ***String: Ingresar*** clic en ***OK***.



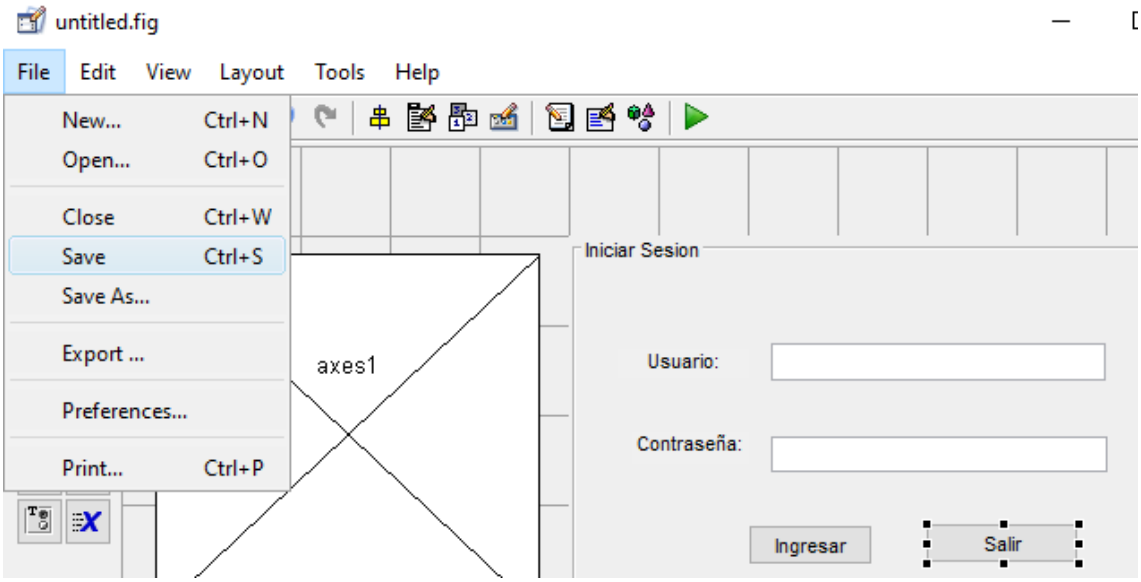
En ***Tag*** digitamos ***btnIngresar***



Los mismos pasos para el siguiente boton digitamos en ***String: Salir*** y en ***Tag: btnSalir***



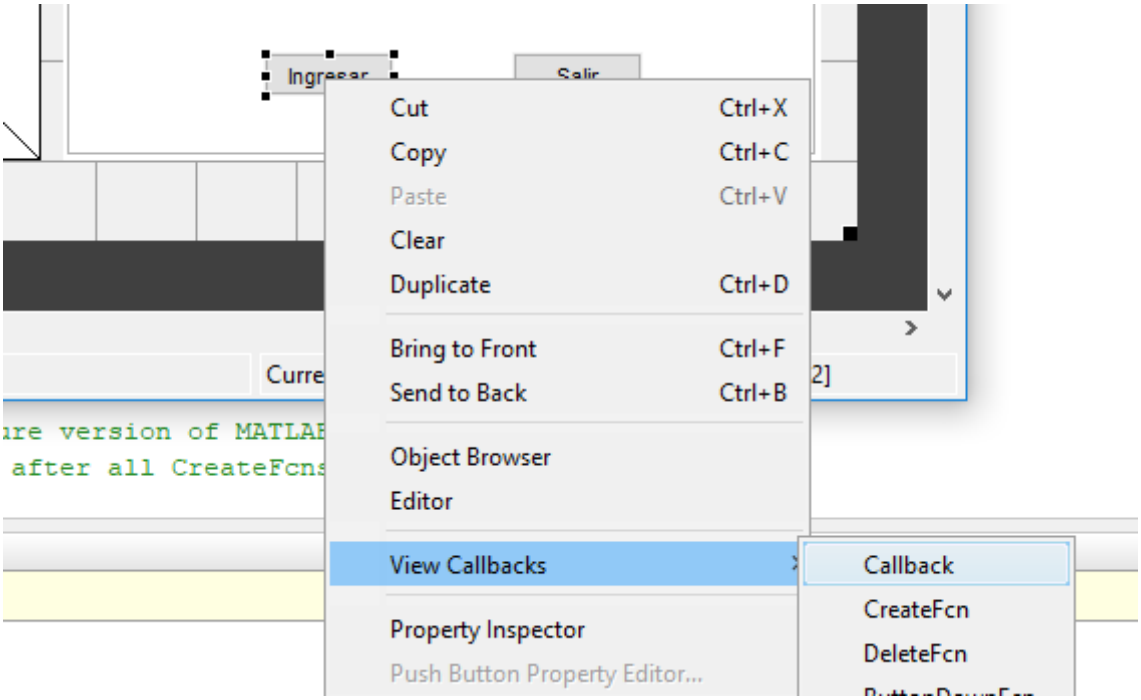
Procedemos a **guardar**



Nombre: frmLogin y guardamos.

CODIFICAMOS EL BOTON INGRESAR:

Para ingresar al codigo hacemos el siguiente paso.



CODIFICAMOS EL BOTON INGRESAR.

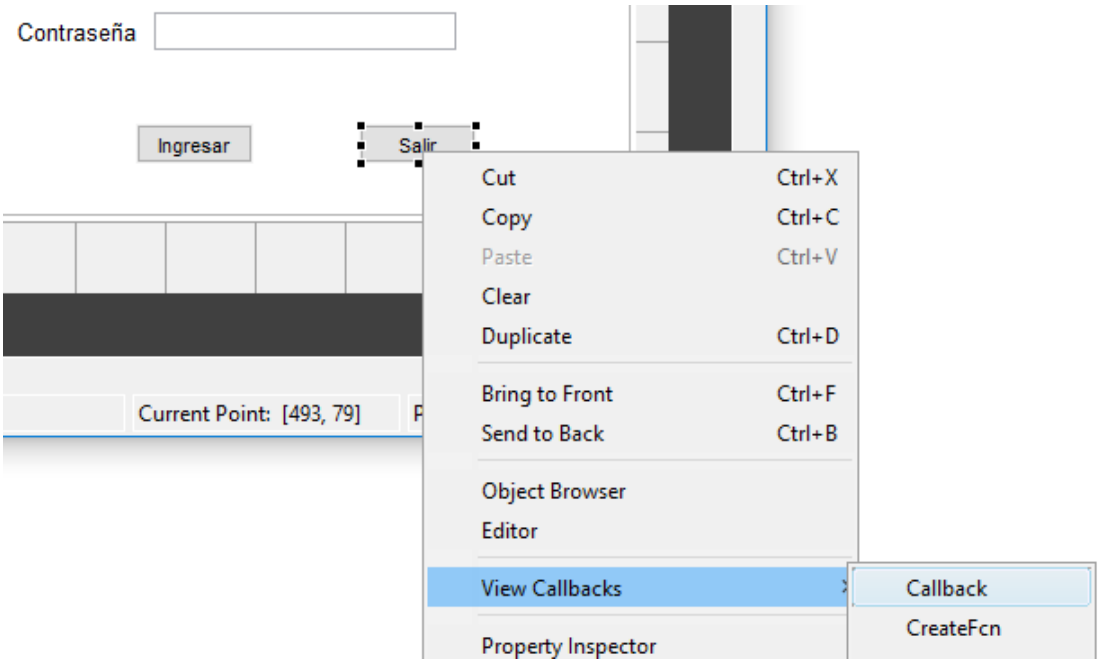
```
119     % --- Executes on button press in btnIngresar.
120     function btnIngresar_Callback(hObject, eventdata, handles)
121     %set(hObject,'String',[get(hObject,'String') '*']);
122
123 -   usuario=(get(handles.txtUsuario,'string'));
124 -   password=(get(handles.txtPassword,'string'));
125
126 -   usu='admin';
127 -   pass='123';
128
129 -   if strcmpi(usuario,usu) && strcmpi(password,pass)
130 -       frmPrincipal;
131 -       close(frmLogin);
132 -   else
133 -       hmsg=warndlg(['El usuario o contraseña son',...
134                     ' Incorrectos, Verifique nuevamente.']);
135 -               pause(2.0);
136 -               delete(hmsg);
137 -   end
138
```

Copia y pega:

```
usuario=(get(handles.txtUsuario,'string'));
password=(get(handles.txtPassword,'string'));

usu='admin';
pass='123';

if strcmpi(usuario,usu) && strcmpi(password,pass)
    frmPrincipal;
    close(frmLogin);
else
    hmsg=warndlg(['El usuario o contraseña son',...
                ' Incorrectos, Verifique nuevamente.']);
          pause(2.0);
          delete(hmsg);
end
```

CODIFICAMOS EL BOTON SALIR:

Para ingresar al codigo hacemos el siguiente paso.



CODIFICAMOS EL BOTON SALIR

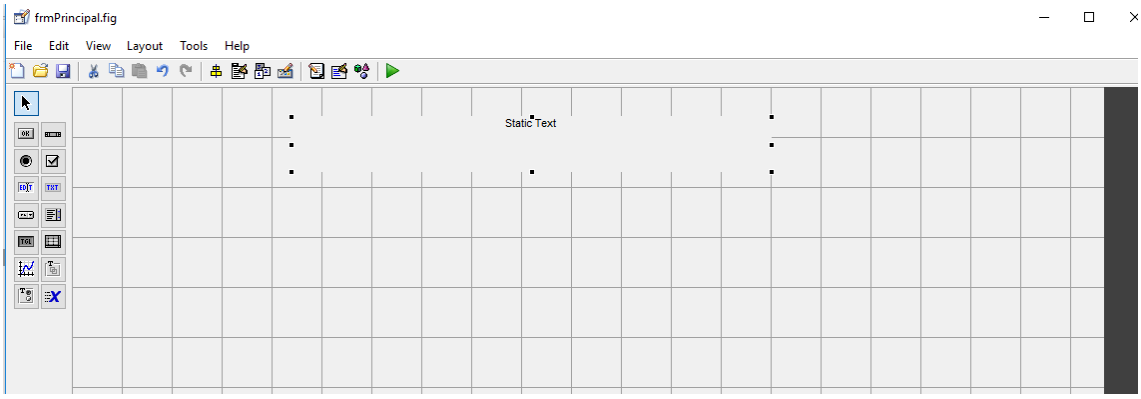
```
138
139     % --- Executes on button press in btnSalir.
140   function btnSalir_Callback(hObject, eventdata, handles)
141 -   close();
142
143
```

Copia y pega:

```
close();
```

Luego creamos una interfaz llamado ***frmPrincipal***

Creamos un **static text**

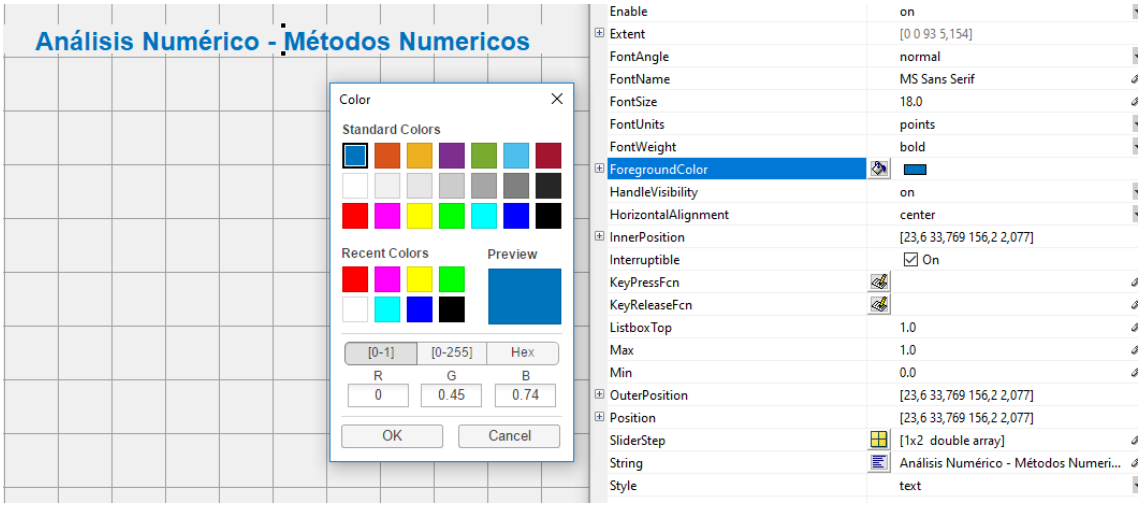


Hacemos doble clic en el ***static text*** y hacemos lo siguiente:

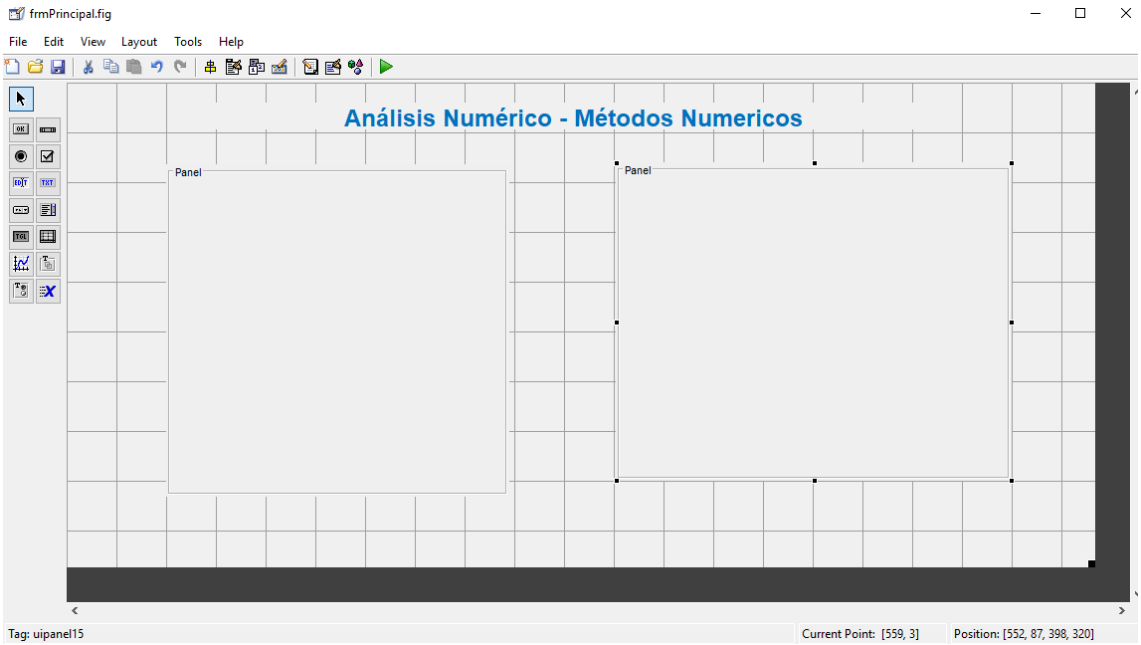
en ***String: Analisis Numerico – Metodos Numericos***.

en ***ForegroundColor:*** cambiamos el color deseado.

En ***Fontsize:*** tamaño de letra deseado.



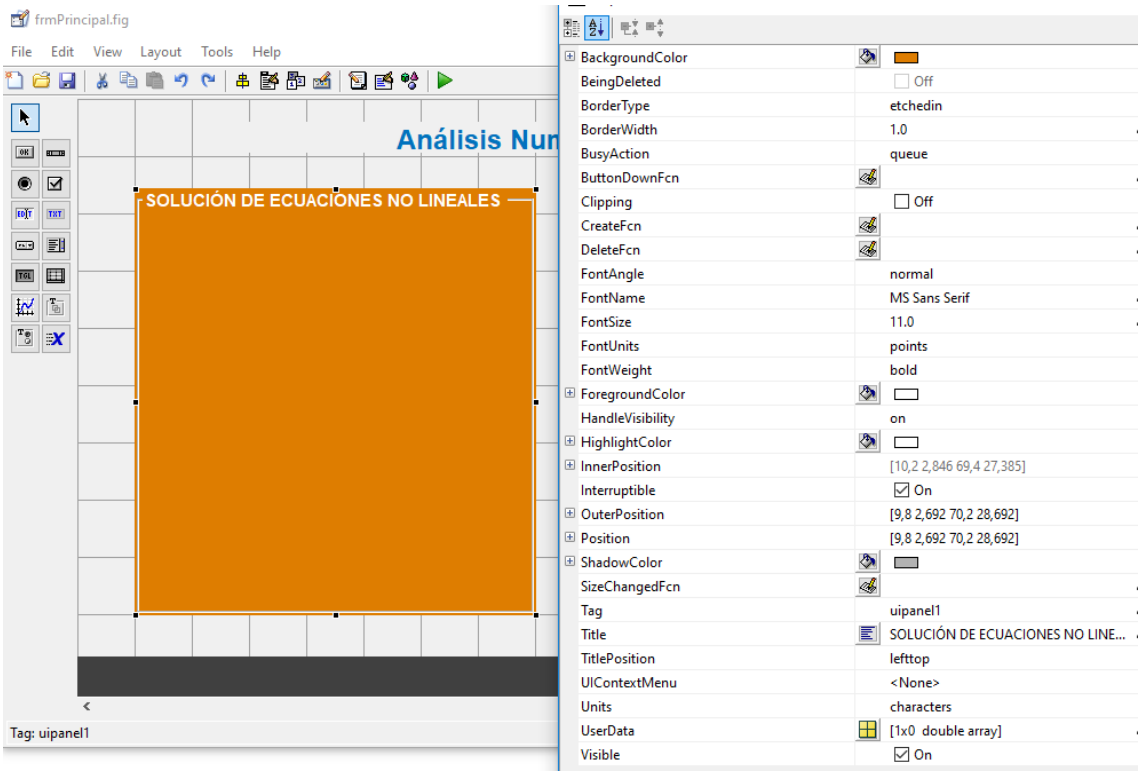
Creamos dos paneles.



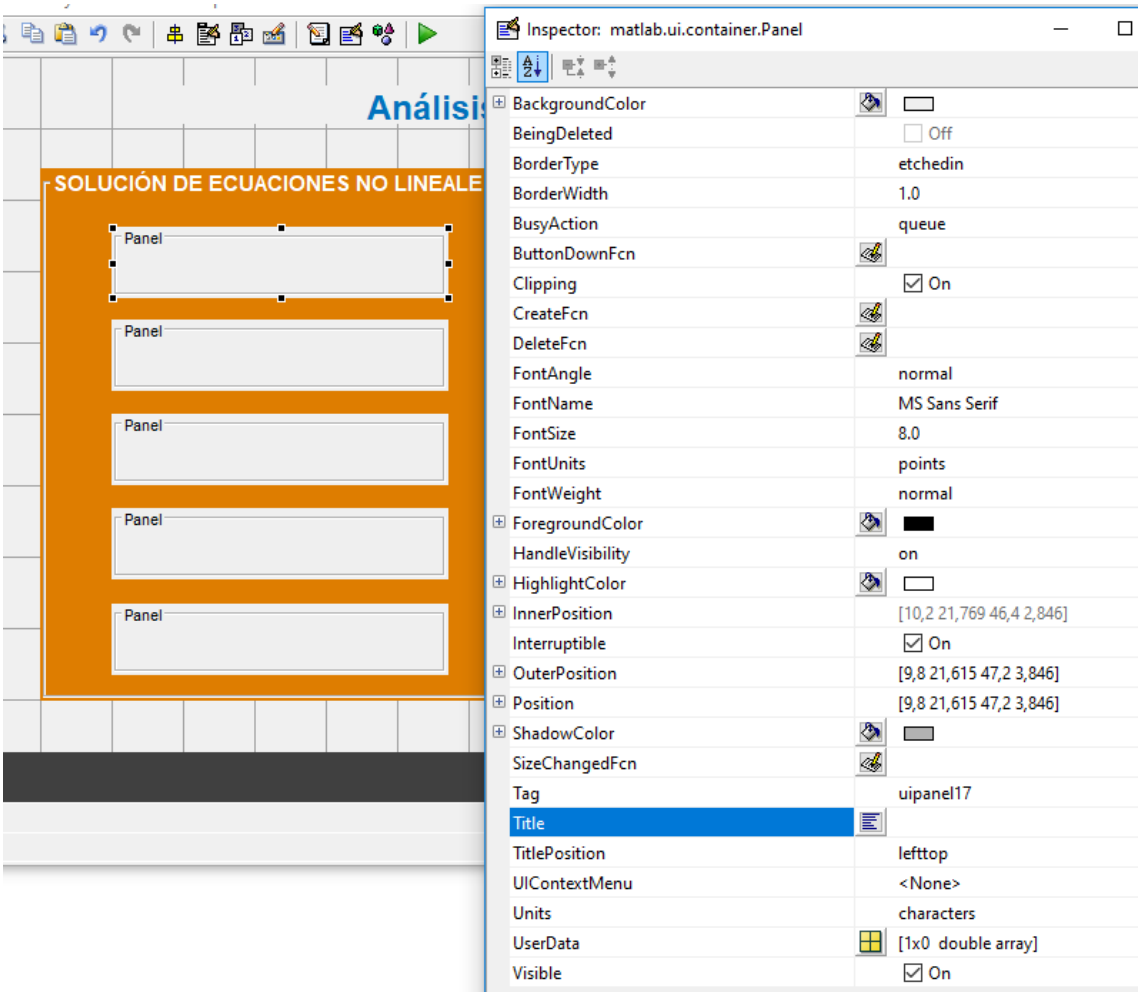
En el primer panel hacemos doble clic y hacemos lo siguiente.

en ***BackgroundColor:*** cambiamos el color deseado.

En ***title: SOLUCIÓN DE ECUACIONES NO LINEALES***



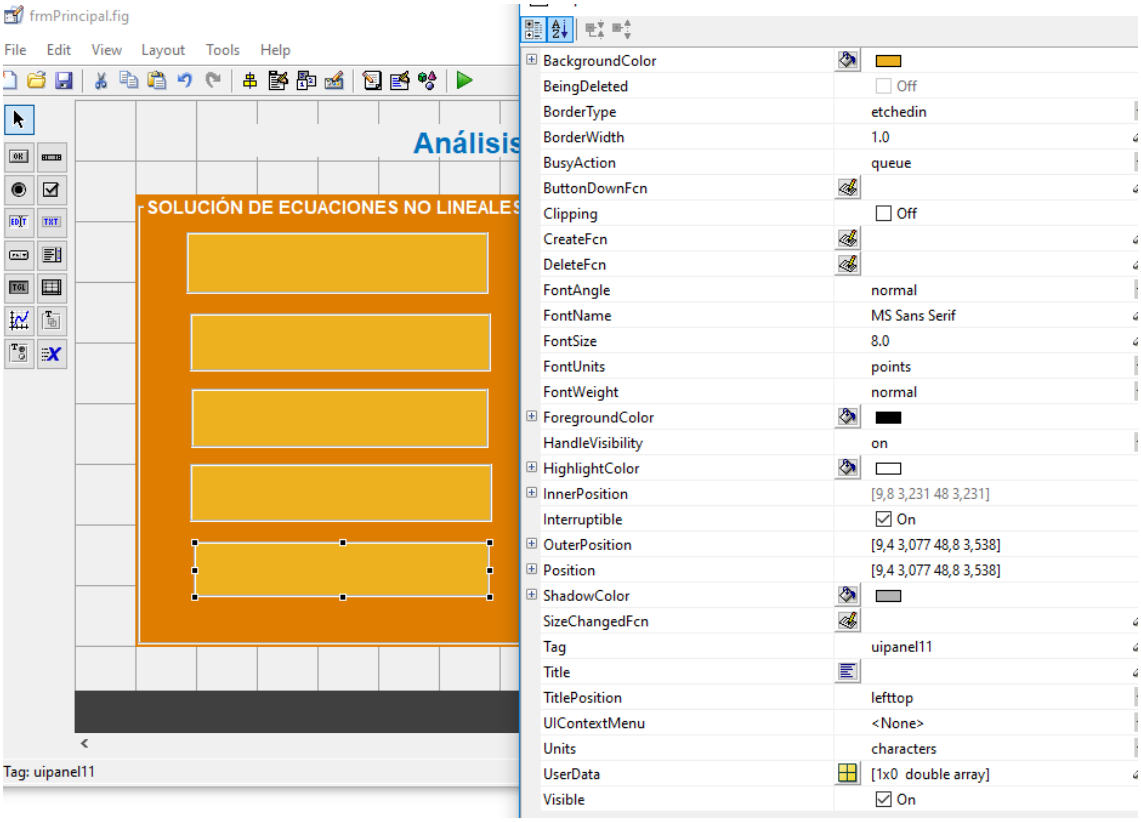
Luego creamos los siguientes paneles luego doble clic en el primero.



en ***BackgroundColor:*** cambiamos el color deseado.

En ***title:*** borramos el texto(queda en blanco).

Lo mismo para todos los paneles y quedaria asi.

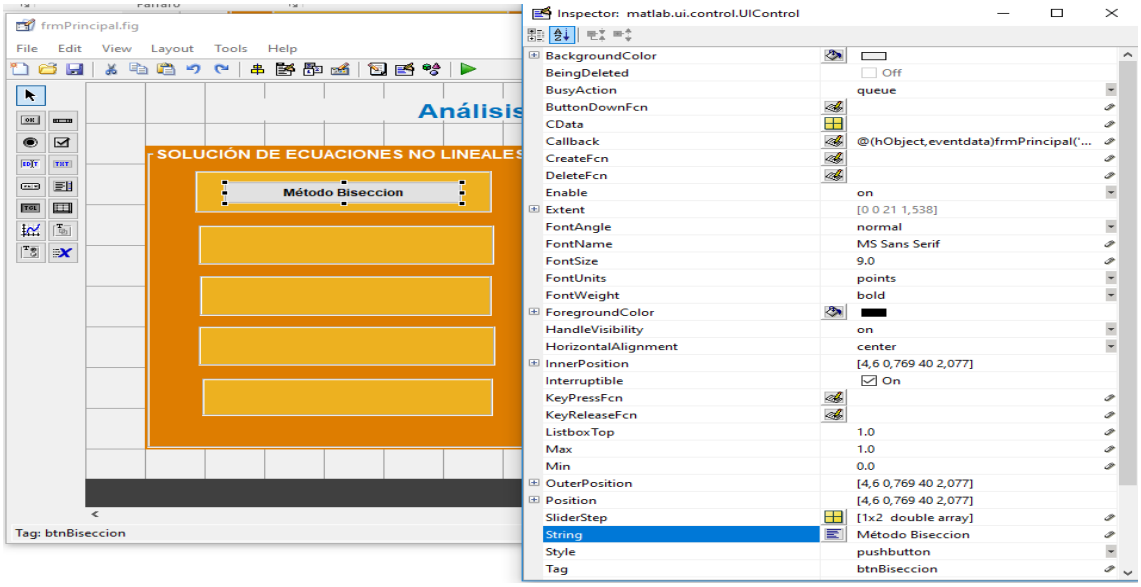


Procedemos a crear el primer boton dentro del panel creado.

Digitamos lo siguiente:

En ***String: Método Biseccion***

En ***Tag: btnBiseccion***



Para los siguientes botones hacer los mismos pasos solo cambiar en ***String*** y en ***Tag*** de la siguiente manera:

Para el segundo boton:

En ***String: Método Falsa Posicion***

En **Tag: btnFalsaPosicion**

Para el tercer boton:

En ***String: Método Punto Fijo***

En ***Tag: btnPuntoFijo***

Para el cuarto boton:

En ***String: Método Newton Rahpson***

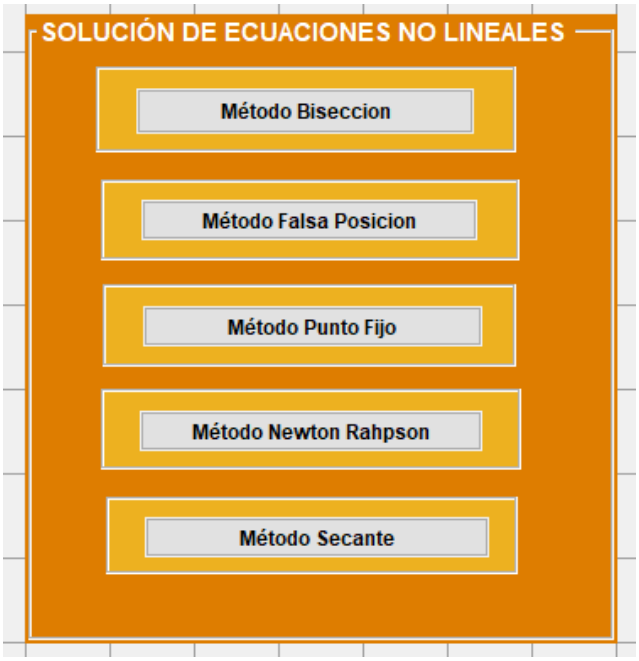
En ***Tag: btnNewtonRahpson***

Para el quinto boton:

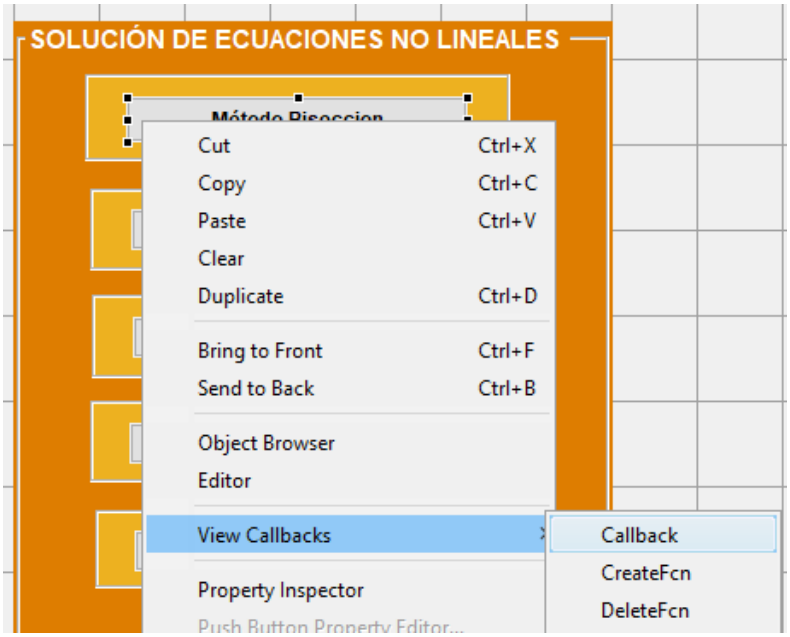
En ***String: Método Secante***

En ***Tag: btnSecante***

Quedaria de esta manera



Hacer este paso para entrar al codigo de todos los botones y hacer la codificacion.



ENLAZAR GUIDES DE ECUACIONES NO LINEALES

Botón Método Biseccion

```
71      % --- Executes on button press in btnBiseccion.
72    function btnBiseccion_Callback(hObject, eventdata, handles)
73 -    frmBiseccion;
```

function btnBiseccion_Callback(hObject, eventdata, handles)

frmBiseccion;

Botón Método Falsa Posicion

```
76      % --- Executes on button press in btnFalsaPosicion.
77    function btnFalsaPosicion_Callback(hObject, eventdata, handles)
78 -    frmFalsaPosicion;
```

function btnFalsaPosicion_Callback(hObject, eventdata, handles)

frmFalsaPosicion;

Botón Método Punto Fijo

```
96      % --- Executes on button press in btnPuntoFijo.
97    function btnPuntoFijo_Callback(hObject, eventdata, handles)
98 -    frmPuntoFijo;
```

function btnPuntoFijo_Callback(hObject, eventdata, handles)

frmPuntoFijo;

Botón Método Newton Raphson

```
91      % --- Executes on button press in btnNewtonRahpson.
92    function btnNewtonRahpson_Callback(hObject, eventdata, handles)
93 -    NewtonRaphson_Guide;
```

function btnNewtonRahpson_Callback(hObject, eventdata, handles)
NewtonRaphson_Guide;

Botón Método de la Secante

```
86      % --- Executes on button press in btnSecante.
87    function btnSecante_Callback(hObject, eventdata, handles)
88 -    frmSecante;
```

function btnSecante_Callback(hObject, eventdata, handles)
frmSecante;

¿COMÓ CREAR SOLUCION DE ECUACIONES POLINOMIALES?

Para el siguiente panel realizar los mismos pasos

Digitamos lo siguiente para los botones:

Para el boton de metodo de muller

En *String: Método Muller*

En *Tag: btnMetodoMuller*

Para el boton de metodo de Bairstow

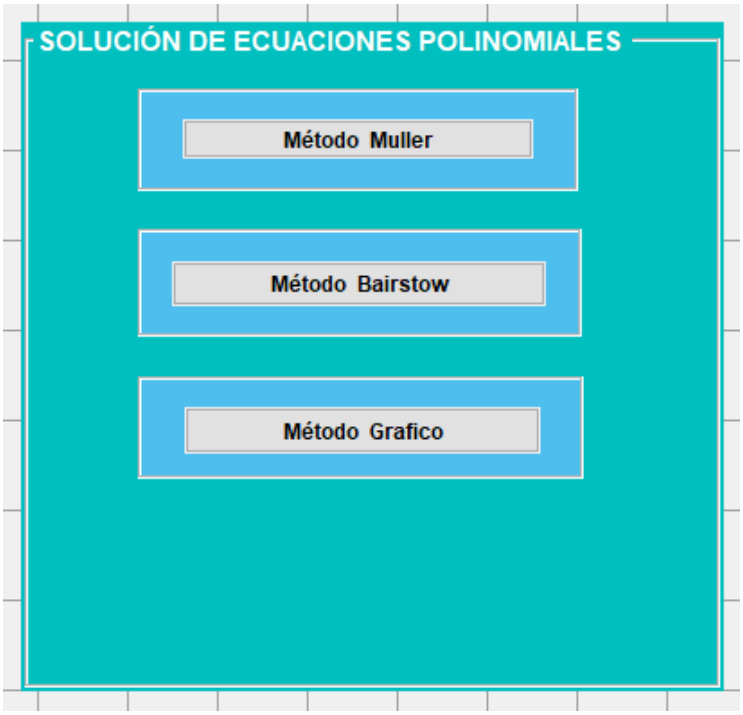
En *String: Método Bairstow*

En *Tag: btnMetodoBairstow*

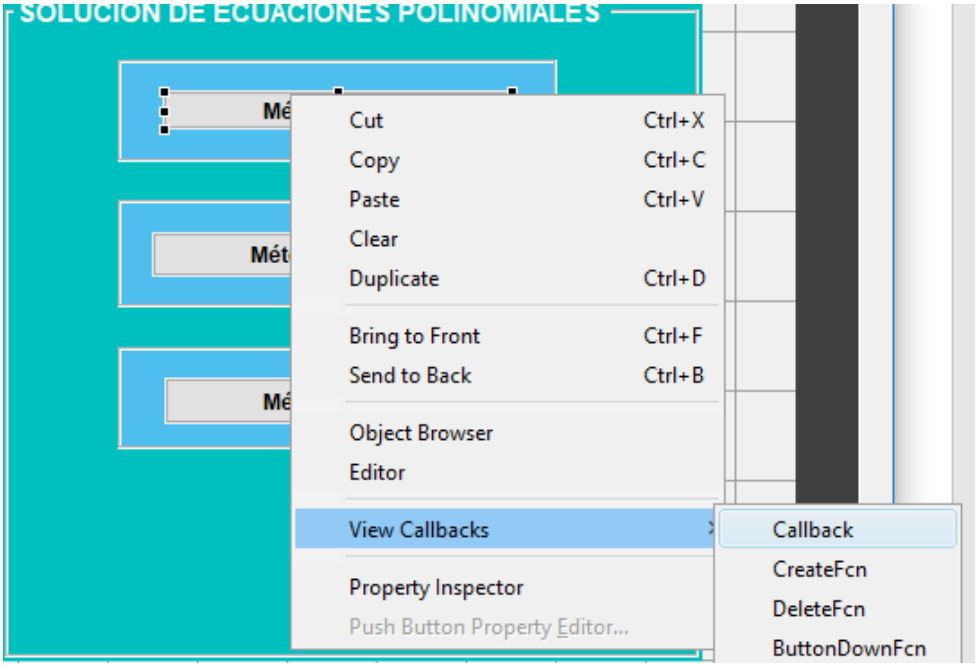
Para el boton de metodo Grafico

En *String: Método Grafico*

En *Tag: btnMetodoGrafico*



Hacer este paso para entrar al codigo de todos los botones y hacer la codificacion.



ENLAZAR GUIDES DE ECUACIONES POLINOMIALES

MÉTODO DE MULLER

```
101     % --- Executes on button press in btnMetodoMuller.
102   function btnMetodoMuller_Callback(hObject, eventdata, handles)
103 -   Muller_Guide;
```

function btnMetodoMuller_Callback(hObject, eventdata, handles)
Muller_Guide;

MÉTODO DE BAIRSTOW

```
106     % --- Executes on button press in btnMetodoBairstow.
107   function btnMetodoBairstow_Callback(hObject, eventdata, handles)
108 -   MetodoBairstow;
```

function btnMetodoBairstow_Callback(hObject, eventdata, handles)
MetodoBairstow;

MÉTODO GRÁFICO

```
81      % --- Executes on button press in btnMetodoGrafico.
82    function btnMetodoGrafico_Callback(hObject, eventdata, handles)
83 -   frmMetodoGrafico;
```

function btnMetodoGrafico_Callback(hObject, eventdata, handles)
frmMetodoGrafico;

Procedemos a crear la interfaz para cada botón

PARA LAS SOLUCIONES NO LINEALES

MÉTODO BISECCION

Creamos 5 statics texts.

- ***Funcion Matematica***
- ***Xi***
- ***Xu***
- ***Error***
- ***La Raiz es***

cada uno con su campo de texto editable (edit text).

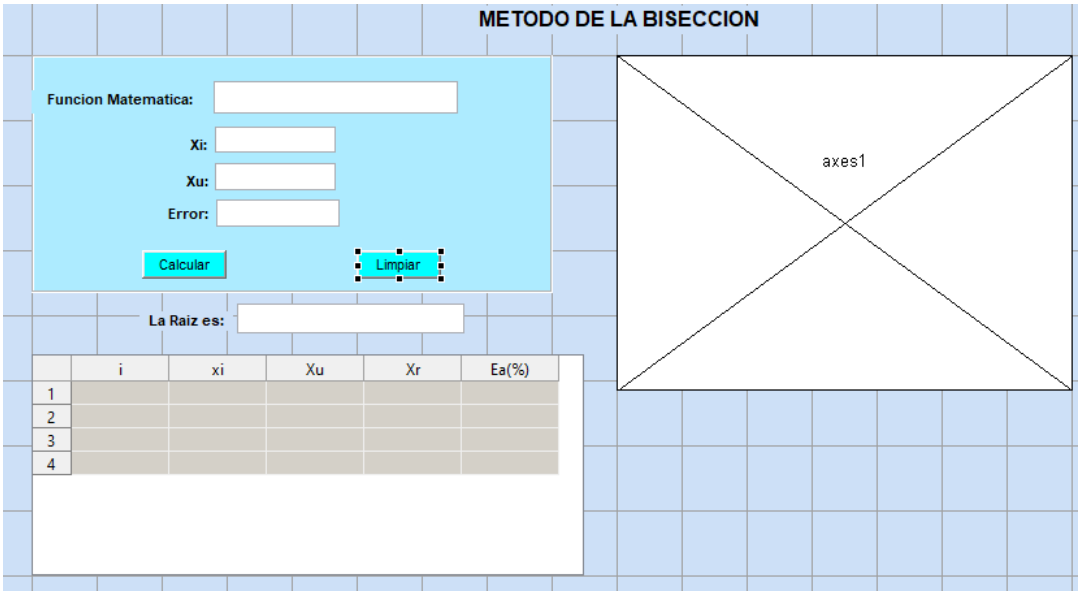
Eliminamos los textos por defecto.

Luego creamos un "***Table***", seguido doble clic y editamos en Tag "***tabla***".

Luego creamos un ***axes1*** para el grafico.

Finalmente creamo dos ***botones*** "***Calcular***" y "***limpiar***", con ***String*** "***calcular***" y ***tag*** "***pushbutton1***", luego el de limpiar seria ***String*** "***Limpiar***" con ***Tag*** "***pushbutton2***".

Y quedaria de la siguiente manera:



CODIFICACIÒN DEL BOTON "CALCULAR".

```
function pushbutton1_Callback(hObject, eventdata, handles)
f=get(handles.edit1,'string');
f=inline(f);
xai=str2double(get(handles.edit2,'string')); % valor de x1
xbi=str2double(get(handles.edit3,'string')); % valor de x2
tol=str2double(get(handles.edit4,'string')); % error
i=1;
ea(1)=100;

if f(xai)*f(xbi)<0; % Comprobando que la raiz se encuentra en este intervalo
xa(1)=xai;
xb(1)=xbi;
xr(1)=(xa(1)+xb(1))/2;
%Limpiar tabla antes de mostrar resultado
set(handles.tabla,'Data',{})
% Limpiar tabla, grafico en caso de que antes se haya graficado una funcion
hold off
cla
```

```
set(handles.tabla,'Data',{})
set(handles.respuesta,'string','No hay raiz');
while abs(ea(i))>=tol
    if f(xa(i))*f(xr(i))<0 % Condicion de cumplimiento
         xa(i+1)=xa(i);
         xb(i+1)=xr(i); % Es la raiz(xr) si se cumple condicion
    end
    if f(xa(i))*f(xr(i))>0 % Condicion de cumplimiento
        xa(i+1)=xr(i); % Es la raiz(xr) si se cumple condicion
        xb(i+1)=xb(i);
    end
  xr(i+1)=(xa(i+1)+xb(i+1))/2; % Valor intermedio para 2° iteracion
  ea(i+1)=abs((xr(i+1)-xr(i))/(xr(i+1))*100);% error absoluto

   % ============= Mostrara datos en tabla ==============================
   newRow ={i xa(i+1) xb(i+1) xr(i+1) ea(i+1)};
   oldData = get(handles.tabla,'Data');
   newData=[oldData; newRow];
   set(handles.tabla,'Data',newData)

   grid on;
   plot(i,f(i),'or');
   hold on;

 i=i+1;
end % Cerramos while
% Mostrando respuesta en textbox con formato coma flotante a 6 cifras decimales
respuesta=sprintf('%0.6f',xr(i));
set(handles.respuesta,'string',respuesta);

%Grafica de la funcion
fplot(handles.axes1,f,[xai xbi]);
%grid on;
%hold on;
%handles.axes1=plot(xr(i),subs(f,respuesta),'r

   grid on;
   plot(xbi,f(xbi),'Ok');
   grid on;
   ezplot(f);

   hold on;
   zoom on
else
set(handles.respuesta,'string','No existe la raiz en el intervalo');
%zoom on
End
```

Ahora para el Botón Limpiar

```
function pushbutton2_Callback(hObject, eventdata, handles)
cla %limpiar tabla
set(handles.tabla,'Data',{})
%limpiar textboxs
set(handles.edit1,'string','');
```

```
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.respuesta,'string','');
```

MÉTODO FALSA POSICIÓN

Creamos 5 statics texts.

- ***Funcion***
- ***Xi***
- ***Xu***
- ***Error***
- ***La Raiz es***

cada uno con su campo de texto editable (edit text).

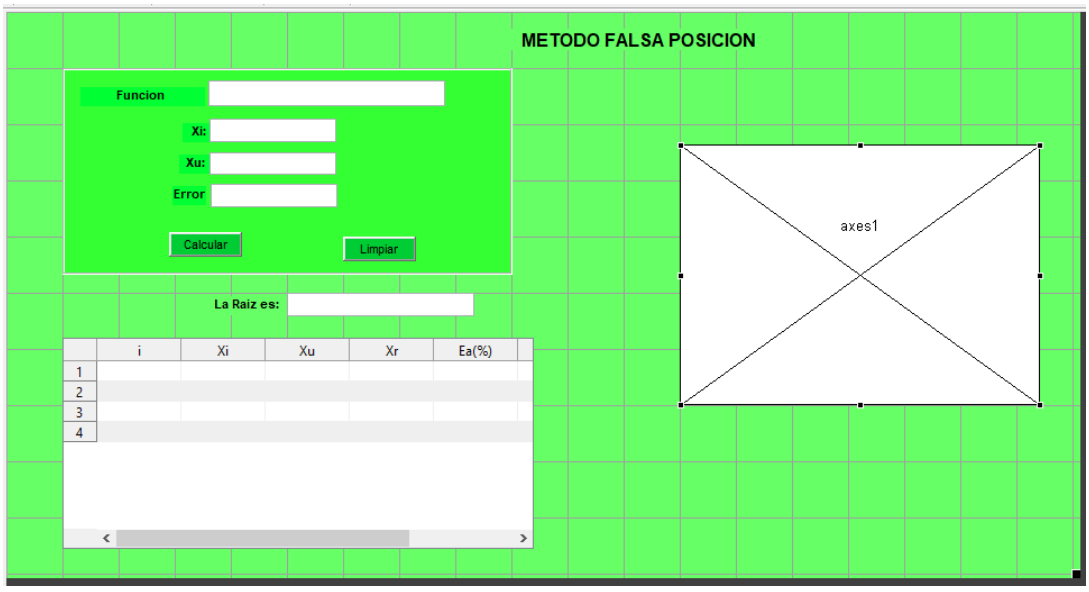
Eliminamos los textos por defecto.

Luego creamos un "***Table***", seguido doble clic y editamos en ***Tag*** "***tabla***".

Luego creamos un ***axes1*** para el grafico.

Finalmente creamo dos botones "***Calcular***" y "***limpiar***", con ***String*** "***calcular***" y ***tag*** "***btnCalcular***", luego el de limpiar serìa ***String*** "***Limpiar***" con ***Tag*** "***btnLimpiar***".

Y quedaria de la siguiente manera:



CODIFICACIÓN DEL BOTON CALCULAR

```
function btnCalcular_Callback(hObject, eventdata, handles)
f=get(handles.edit1,'string');
f=inline(f);
xai=str2double(get(handles.edit2,'string')); % valor de x1
xbi=str2double(get(handles.edit3,'string')); % valor de x2
tol=str2double(get(handles.edit4,'string')); % error
i=1;
ea(1)=100;
%%%%% Metodo Bisección %%%%%%
if f(xai)*f(xbi)<0; % Comprobando que la raiz se encuentra en este intervalo
xa(1)=xai;
xb(1)=xbi;
xf(1)= xb(1)-f(xb(1))*(xa(1) - xb(1))/(f(xa(1))-f(xb(1)));
%Limpiar tabla antes de mostrar resultado
set(handles.tabla,'Data',{})
% Limpiar tabla, grafico en caso de que antes se haya graficado una funcion
hold off
cla
set(handles.tabla,'Data',{})
set(handles.respuesta,'string','No hay raiz');
while abs(ea(i))>=tol
if f(xa(i))*f(xf(i))<0 % Condicion de cumplimiento
xa(i+1)=xa(i);
xb(i+1)=xf(i); % Es la raiz(xr) si se cumple condicion
end
if f(xa(i))*f(xf(i))>0 % Condicion de cumplimiento
xa(i+1)=xf(i); % Es la raiz(xr) si se cumple condicion
xb(i+1)=xb(i);
end
xf(i+1)= xb(i+1)-f(xb(i+1))*(xa(i+1) - xb(i+1))/(f(xa(i+1))-f(xb(i+1)));
ea(i+1)=abs((xf(i+1)-xf(i))/(xf(i+1))*100);% error absoluto
% Mostrara datos en tabla
   %valores = {i xa(i+1) xb(i+1) xf(i+1) ea(i+1)};
   %temp=get(handles.tabla,'data');
   %valoresNuevos=[valores;temp];
   %set(handles.tabla,'Data',valoresNuevos)

newRow ={i xa(i+1) xb(i+1) xf(i+1) ea(i+1)};
    oldData = get(handles.tabla,'Data');
    newData=[oldData; newRow];
    set(handles.tabla,'Data',newData)

    grid on;
    plot(i,f(i),'or');
    hold on;

i=i+1;
end % Cerramos while
% Mostrando respuesta en textbox con formato coma flotante a 6 cifras decimales
respuesta=sprintf('%0.6f',xf(i));
set(handles.respuesta,'string',respuesta);
%axes1 de la funcion
```

```
fplot(handles.axes1,f,[xai xbi]);
%grid on;
%hold on;
%handles.axes1=plot(xf(i),subs(f,respuesta),'r
    grid on;
    plot(xbi,f(xbi),'Ok');
    grid on;
    ezplot(f);

    hold on;
    zoom on
else
set(handles.respuesta,'string','No existe la raiz en el intervalo');

end
```

Para el boton Limpiar

```
function btnLimpiar_Callback(hObject, eventdata, handles)
cla %limpiar tabla
set(handles.tabla,'Data',{})
%limpiar textboxs
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.respuesta,'string','');
```

MÉTODO PUNTO FIJO

Creamos 4 statics texts.

- ***Funcion***
- ***Funcion Despejada***
- ***Xo***
- ***Tolerancia***

cada uno con su campo de texto editable (***edit text***).

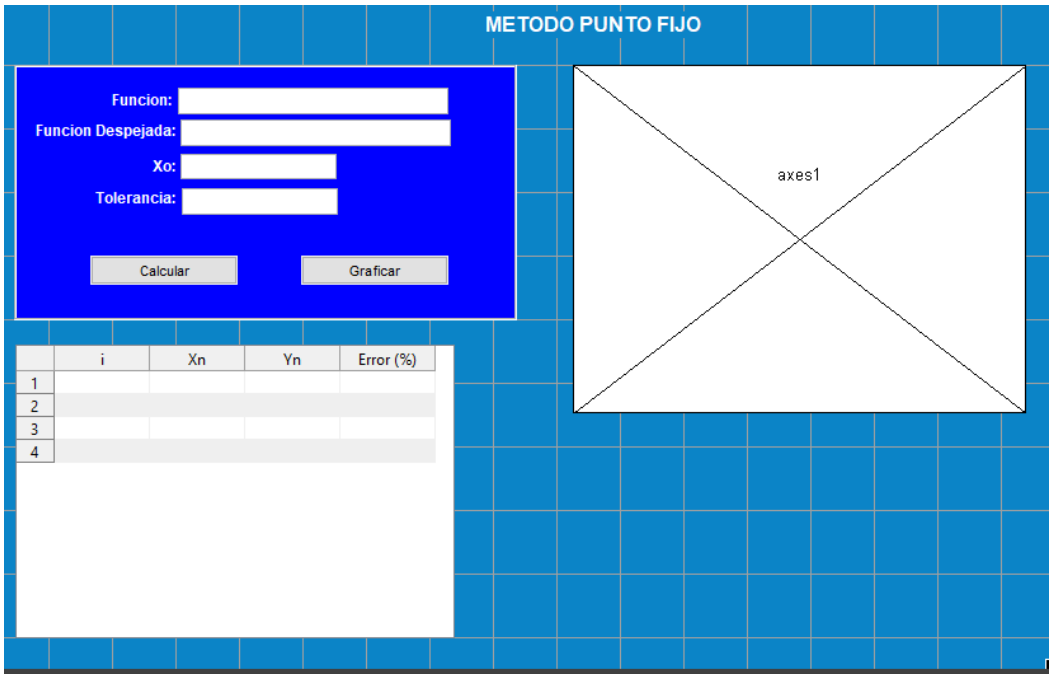
Eliminamos los textos por defecto.

Luego creamos un "***Table***", seguido doble clic y editamos en ***Tag*** "***tabla***".

Luego creamos un ***axes1*** para el grafico.

Finalmente creamo dos botones "***Calcular***" y "***Graficar***", con ***String*** "***calcular***" y ***tag*** "***btnCalcular***", luego el de limpiar seria ***String*** "***Limpiar***" con ***Tag*** "***btnLimpiar***".

Y quedaria de la siguiente manera:



CODIFICACIÓN DEL BOTON CALCULAR

```
function btnCalcular_Callback(hObject, eventdata, handles)

f=inline(get(handles.txtFuncion,'string'));
g=inline(get(handles.txtFuncionD,'string'));
Xo=str2double(get(handles.txtValorInicial,'string'));
Tol=str2double(get(handles.txtTolerancia,'string'));

Iter=10;

Yn=f(Xo);
Error=Tol+1;
Cont=0;

%Z es una matriz la cual permitira observar lo datos como una tabla a %la finalizacion del
programa
%La sentencia While ejecuta todas las órdenes mientras la expresión %sea verdadera.
Z=[Cont,Xo,Yn,Error];
set(handles.uitable1,'Data',[]);
while Yn~=0 && Error>Tol && Cont<Iter
Xn=g(Xo);
Yn=f(Xn);
Error=abs((Xn-Xo)/Xn);
%Error=abs(Xn-X0);
Cont=Cont+1;
Z(Cont,1)=Cont;
Z(Cont,2)=Xn;
Z(Cont,3)=Yn;
Z(Cont,4)=Error;
%las z son las posiciones asignadas en la tabla a los resultados que %se observarán
```

```
Xo=Xn;

%para agregar a la tabla de valores hallados
    newRow ={Cont,Xn,Yn,Error};
    oldData = get(handles.uitable1,'Data');
    newData=[oldData; newRow];
    set(handles.uitable1,'Data',newData)

end
%La sentencia if tiene como función evaluar condiciones, que en caso
%de ser verdadera se procede a realizar ciertos pasos, de lo contrario
%se procede a realizar otros, por medio de la funcion else.
if Yn==0
     fprintf('\n\nSOLUCION:\n')
      fprintf('%g es raiz\n\n',Xo);
else
   if Error<Tol
     fprintf('\n\nSOLUCION:\n')
     fprintf('%g es una aproximacion con un tolerancia de %g\n\n',Xo,Tol);
   end
end
```

Para el botón Graficar

```
function btnGraficar_Callback(hObject, eventdata, handles)
% hObject    handle to btnGraficar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.axes1)
f=inline(get(handles.txtFuncion,'string'));
ezplot(f);
```

MÉTODO NEWTON RAPHSON

Creamos 3 statics texts.

- ***Funcion***
- ***Xo***
- ***Error***
- ***La raìz es***

cada uno con su campo de texto editable (***edit text***).

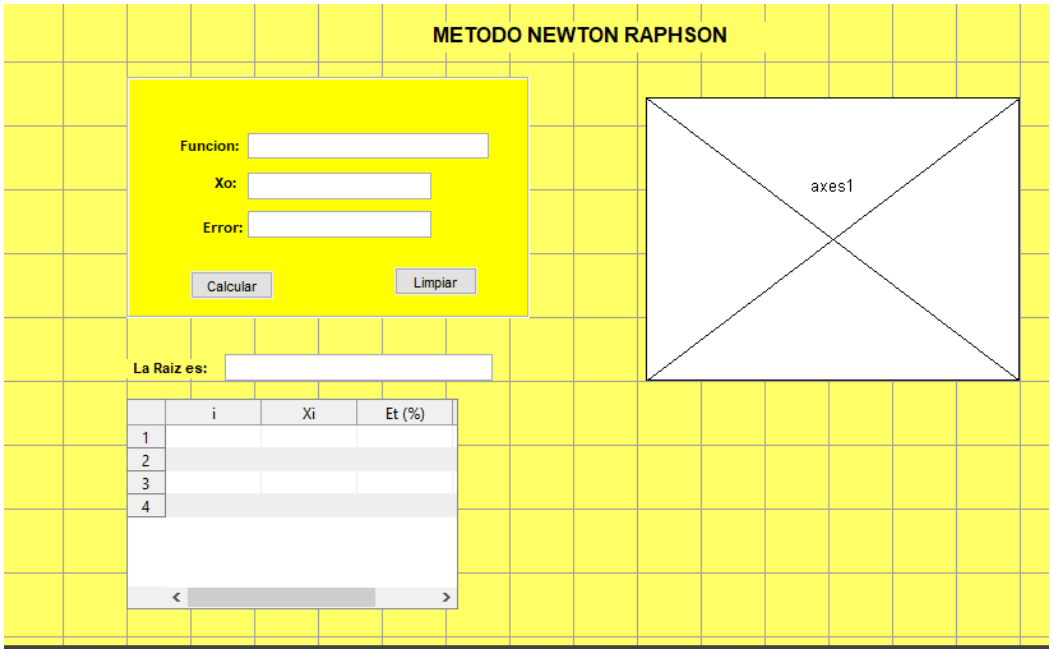
Eliminamos los textos por defecto.

Luego creamos un "***Table***", seguido doble clic y editamos en ***Tag*** "***tabla***".

Luego creamos un ***axes1*** para el grafico.

Finalmente creamo dos botones "*Calcular*" y "*Graficar*", con ***String*** "*calcular*" y ***tag*** "*btnCalcular*", luego el de limpiar seria ***String*** "*Limpiar*" con ***Tag*** "*btnLimpiar*".

Y quedaria de la siguiente manera:



CODIFICACIÓN DEL BOTON CALCULAR

```
function btnCalcular_Callback(hObject, eventdata, handles)
%manejo de excepciones con try y catch
try
funcion=get(handles.txtFuncion,'string');
x0=str2double(get(handles.txtX0,'string'));
porcentajeError=str2double(get(handles.txtPorcentajeError,'string'));
syms x
iteracion=0;
errorCalculado=100;
f=sym(funcion);
derivada=diff(f,x);
%Limpiar tabla antes de mostrar resultado
set(handles.tabla,'Data',{})
%Comprobando que la derivada no sea cero.Casocontrario mostrar un mensaje
%que no hay raiz p.e se ingreso uan constante en lugar de una funcion de x
if derivada==0
    %Limpiar tabla, grafico en caso de que antes se haya graficado uan
    %funcion
      hold off
      cla
      set(handles.tabla,'Data',{})
      set(handles.txtRaiz,'string','No hay raiz');
else
%Iteraciones sucesivas para una mejor aproximacion de la raiz por el metodo
%N-R
while errorCalculado>porcentajeError
```

```
    fx=subs(f,x0);
    dx=subs(derivada,x0);
    x1=x0-(fx/dx);
    errorCalculado=abs(((x1-x0)/x1)*100);
     %mostrar datos en tabla

   %valores = {iteracion x0 x1 errorCalculado};
    %temp=get(handles.tabla,'data');
    %valoresNuevos=[valores;temp];
    %set(handles.tabla,'Data',valoresNuevos)

    newRow ={iteracion x0 errorCalculado};
    oldData = get(handles.tabla,'Data');
    newData=[oldData; newRow];
    set(handles.tabla,'Data',newData)

    x0=x1;
    iteracion=iteracion+1;
end
%Mostrando respuesta en textbox con formato coma flotante a 16 cifras decimales
    respuesta=sprintf('%0.16f',x1);
    set(handles.txtRaiz,'string',respuesta);
%Grafica de la funcion
hold off
handles.axes1=ezplot(f);
grid on;
hold on;
%handles.axes1=plot(x1,subs(f,respuesta),'r*');
zoom on
end
catch
      % msgbox('Un error ha ocurrido. Verifique que ha introducido todos los datos y de la
forma adecuada','Error','error')
End
```

Para el boton Limpiar

```
function btnLimpiar_Callback(hObject, eventdata, handles)
%limpiar area de grafico
cla
%limpiar tabla
set(handles.tabla,'Data',{})
%limpiar textboxs
set(handles.txtFuncion,'string','');
set(handles.txtX0,'string','');
set(handles.txtPorcentajeError,'string','');
set(handles.txtRaiz,'string','');
```

MÉTODO DE LA SECANTE

Creamos 5 statics texts.

- Funcion
- Xi-1
- Xi
- Error
- La raiz es

cada uno con su campo de texto editable (***edit text***).

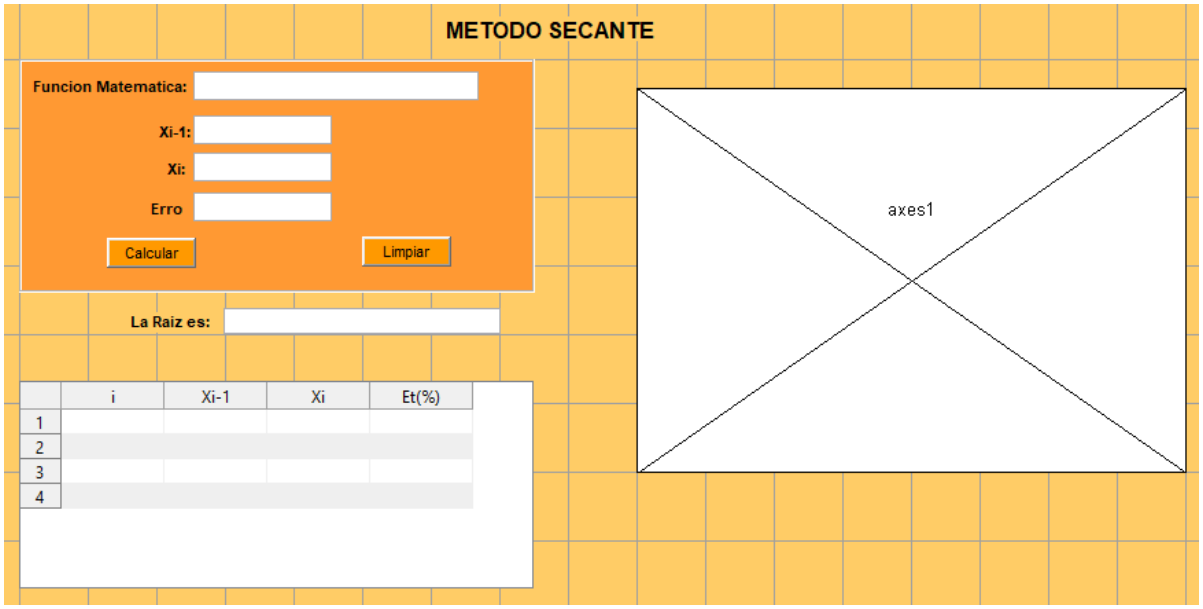
Eliminamos los textos por defecto.

Luego creamos un "***Table***", seguido doble clic y editamos en ***Tag*** "***tabla***".

Luego creamos un ***axes1*** para el grafico.

Finalmente creamo dos botones "***Calcular***" y "***Graficar***", con ***String*** "***calcular***" y ***tag*** "***btnCalcular***", luego el de limpiar seria ***String*** "***Limpiar***" con ***Tag*** "***btnLimpiar***".

Y quedaria de la siguiente manera:



CODIFICACIÓN DEL BOTON CALCULAR

```
function btnCalcular_Callback(hObject, eventdata, handles)
f=get(handles.edit1,'string');
x0=str2double(get(handles.edit2,'string'));%
x1=str2double(get(handles.edit3,'string'));
tol=str2double(get(handles.edit4,'string'));
syms x;
ea(1)=100;
```

```
%Limpiar tabla antes de mostrar resultado
set(handles.tabla,'Data',{})
%Limpiar tabla, grafico en caso de que antes se haya graficado una funcion
hold off
cla
set(handles.tabla,'Data',{})
set(handles.respuesta,'string','No hay raiz');
i=1;
while abs(ea)>tol;
x=x0;
g=eval(f);
x=x1;
gg=eval(f);
xi=x1-((gg*(x0-x1))/(g-gg));
ea=abs((xi-x1)/xi)*100;
x0=x1;
x1=xi;
% Mostrara datos en tabla
    %valores = {i,x xi,ea};
    %temp=get(handles.tabla,'data');
    %valoresNuevos=[valores;temp];
    %set(handles.tabla,'Data',valoresNuevos)
   newRow ={i,x xi,ea};
   oldData = get(handles.tabla,'Data');
   newData=[oldData; newRow];
   set(handles.tabla,'Data',newData)
i=i+1;
end
%Mostrando respuesta en textbox con formato coma flotante a 6 cifras decimales
respuesta=sprintf('%0.6f',xi);
set(handles.respuesta,'string',respuesta);
%Grafica de la funcion
hold off
fplot(handles.axes1,f,[0 xi+1]);
grid on;
hold on;
%handles.axes1=plot(xi,subs(f,respuesta),'r*');
zoom on
% --- Executes on button press in btnLimpiar.
function btnLimpiar_Callback(hObject, eventdata, handles)
cla %limpiar tabla
set(handles.tabla,'Data',{})
%limpiar textboxs
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.respuesta,'string','');
```

Para el botón Limpiar

```
function btnLimpiar_Callback(hObject, eventdata, handles)
cla %limpiar tabla
set(handles.tabla,'Data',{})
%limpiar textboxs
set(handles.edit1,'string','');
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.respuesta,'string','');
```

SOLUCIONES POLINOMIALES

MÉTODO DE MULLER

Creamos 5 statics texts.

- ***Ingrese la Funcion***
- ***Ingrese Xo***
- ***Ingrese Xi***
- ***Ingrese X2***
- ***Ingrese ea***

cada uno con su campo de texto editable (***edit text***).

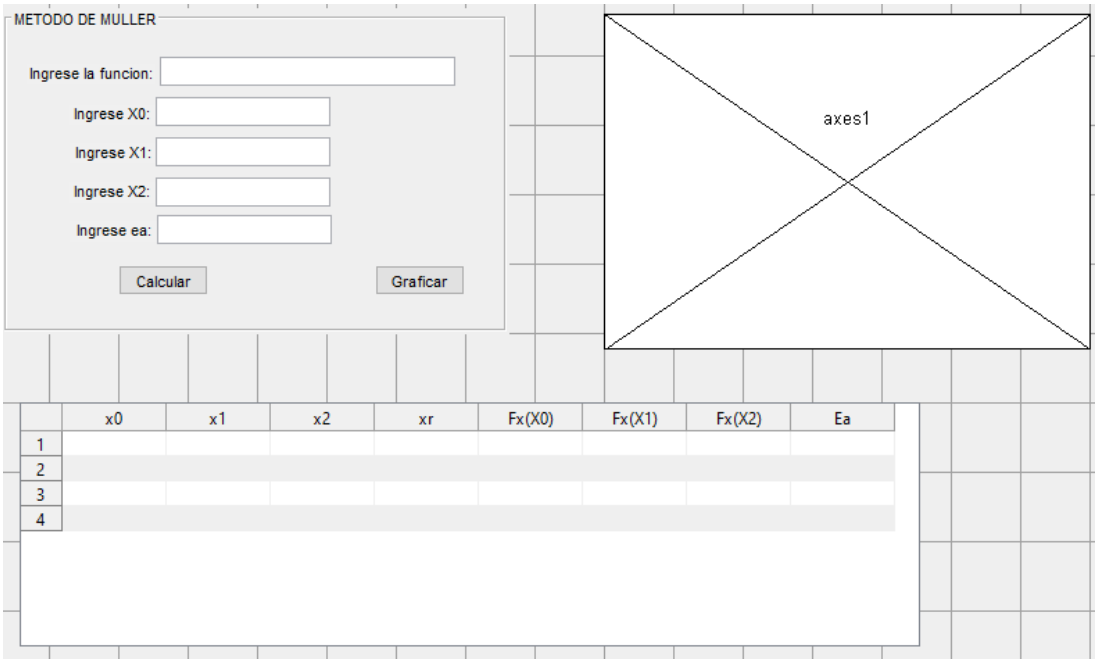
Eliminamos los textos por defecto.

Luego creamos un "***Table***", seguido doble clic y editamos en ***Tag*** "***tabla***".

Luego creamos un ***axes1*** para el grafico.

Finalmente creamo dos botones "***Calcular***" y "***Graficar***", con ***String*** "***calcular***" y ***tag*** "***btnCalcular***", luego el de limpiar seria ***String*** "***Limpiar***" con ***Tag*** "***btnLimpiar***".

Y quedaria de la siguiente manera:



CODIFICACIÓN DEL BOTON CALCULAR

```
function btnCalcular_Callback(hObject, eventdata, handles)
funcion=inline(get(handles.txtF,'string'));
x0=str2double(get(handles.txtX0,'string'));
x1=str2double(get(handles.txtX1,'string'));
x2=str2double(get(handles.txtX2,'string'));
e=str2double(get(handles.txtEa,'string'));

k=0;
xi=0;
sigue=1;
set(handles.uitable1,'Data',[]);
%-----------------------------------------------%
while(sigue)
    xi=x2;
    k=k+1;

    h0=x1-x0;
    h1=x2-x1;
    d0=(funcion(x1)-funcion(x0))/h0;
    d1=(funcion(x2)-funcion(x1))/h1;
    %hallando las constantes
    a=(d1-d0)/(h1+h0);
    b=a*h1+d1;
    c=funcion(x2);

    raizd=sqrt(b*b-4*a*c);
    %-------------------------------------------------%
    if abs (b+raizd)> abs(b-raizd)
        den=b+raizd;
    else
        den=b-raizd;
```

```
    end
    %-------------------------------------------------%

    dxr=-2*c/den;
    xr=x2+dxr;
    sigue=abs(dxr)/xr>e || k<c || abs(funcion(xr))>e;
    et=abs(xr-x2)/xr;
    ea=et*100;

    %para agregar a la tabla de valores hallados
    newRow ={x0,x1,x2,xr,funcion(x0),funcion(x1),funcion(x2),ea};
    oldData = get(handles.uitable1,'Data');
    newData=[oldData; newRow];
    set(handles.uitable1,'Data',newData)

    x0=x1;
    x1=x2;
    x2=xr;
end
```

Para el boton Gráficar

```
function btnGraficar_Callback(hObject, eventdata, handles)
axes(handles.axes1)
funcion=inline(get(handles.txtF,'string'));
ezplot(funcion);
```

MÉTODO BAIRSTOW

Creamos 7 statics texts.

- ***Coeficientes de Funcion***
- ***P***
- ***Q***
- ***Error***
- X_1
- X_2
- ***Ingrese funcion***

cada uno con su campo de texto editable (***edit text***).

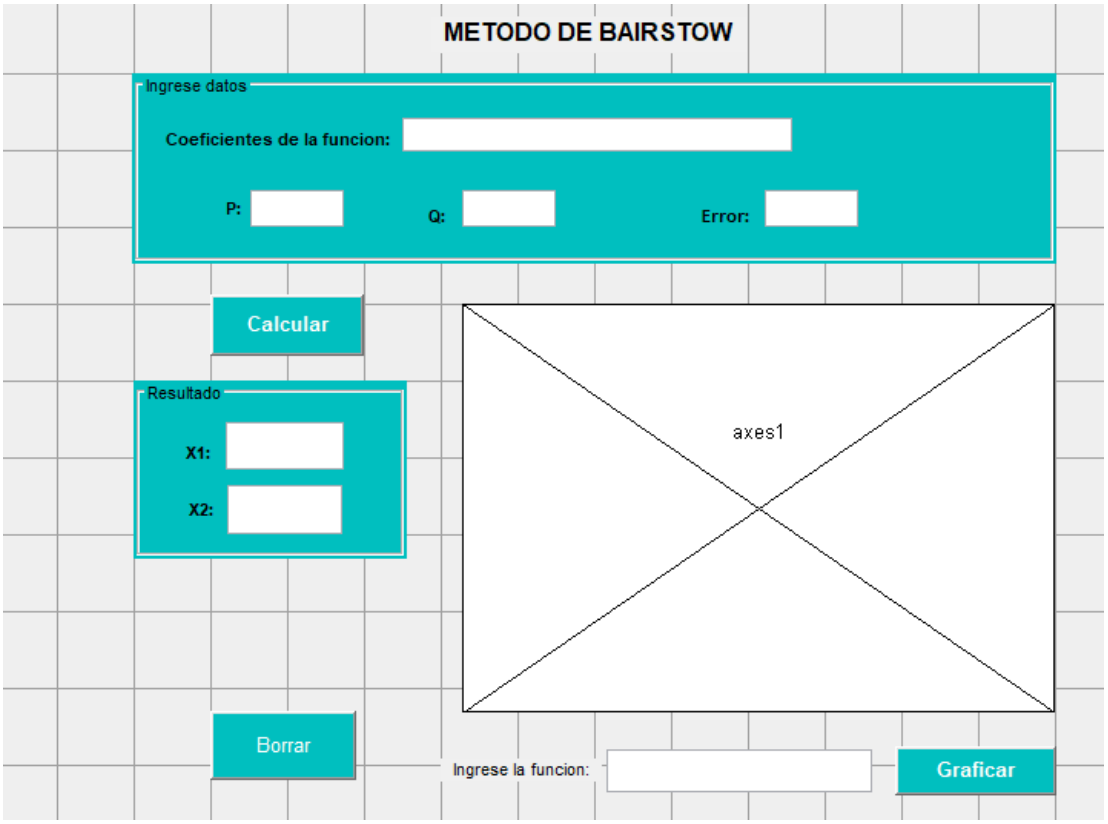
Eliminamos los textos por defecto.

Luego creamos un "***Table***", seguido doble clic y editamos en ***Tag*** "***tabla***".

Luego creamos un ***axes1*** para el grafico.

Finalmente creamo dos botones "*Calcular*" y "*Graficar*", con ***String*** "*calcular*" y ***tag*** "*btnCalcular*", luego el de limpiar seria ***String*** "*Limpiar*" con ***Tag*** "*btnLimpiar*".

Y quedaria de la siguiente manera:



CODIFICACIÓN DEL BOTON CALCULAR

```
function btnCalcular_Callback(hObject, eventdata, handles)
a=str2num(get(handles.edit1,'string'));
p=str2double(get(handles.edit2,'string'));
q=str2double(get(handles.edit3,'string'));
E=str2double(get(handles.edit4,'string'));
n=length(a);
for k=n:-1:1
    b(n+1)=0;
    b(n+2)=0;
    b(k)=a(k)-p*b(k+1)-q*b(k+2);
end
for i=n:-1:1
    c(n+1)=0;
    c(n+2)=0;
    c(i)=b(i)-p*c(i+1)-q*b(i+2);
end
P=(b(1)*c(4)-b(2)*c(3))/(c(2)*c(4)-(c(3))^2);
Q=(b(2)*c(2)-b(1)*c(3))/(c(2)*c(4)-(c(3))^2);
while P>E & Q>E
    p=p+P;
    q=q+Q;
```

```
    for k=n:-1:1
        b(k)=a(k)-p*b(k+1)-q*b(k+2);
        b(n+1)=0;
        b(n+2)=0;
    end
    for i=n:-1:1
        c(i)=b(i)-p*c(i+1)-q*b(i+2);
        c(n+1)=0;
        c(n+2)=0;
    end
      P=(b(1)*c(4)-b(2)*c(3))/(c(2)*c(4)-(c(3))^2);
      Q=(b(2)*c(2)-b(1)*c(3))/(c(2)*c(4)-(c(3))^2);
end
    p=p+P;
    q=q+Q;
    X1=(-p+sqrt(p^2-4*q))/2;
    X2=(-p-sqrt(p^2-4*q))/2;
    set(handles.edit5,'string',X1);
    set(handles.edit6,'string',X2);
```

Para el boton Gráficar

```
function btnGraficar_Callback(hObject, eventdata, handles)
f=get(handles.edit7,'string');
f=inline(f)
ezplot(f),grid on
```

Para el boton Borrar

```
function btnBorrar_Callback(hObject, eventdata, handles)
set(handles.edit1,'string','')
set(handles.edit2,'string','')
set(handles.edit3,'string','')
set(handles.edit4,'string','')
set(handles.edit5,'string','')
set(handles.edit6,'string','')
set(handles.edit7,'string','ingrese la funcion a graficar')
set(handles.edit7,'visible','off')
set(handles.axes1,'Visible','off')
```

MÉTODO GRÁFICO

Creamos 4 statics texts.

- ***Funcion 1***
- X_1
- ***Funcion 2***
- X_2

Cada uno con su campo de texto editable (***edit text***).

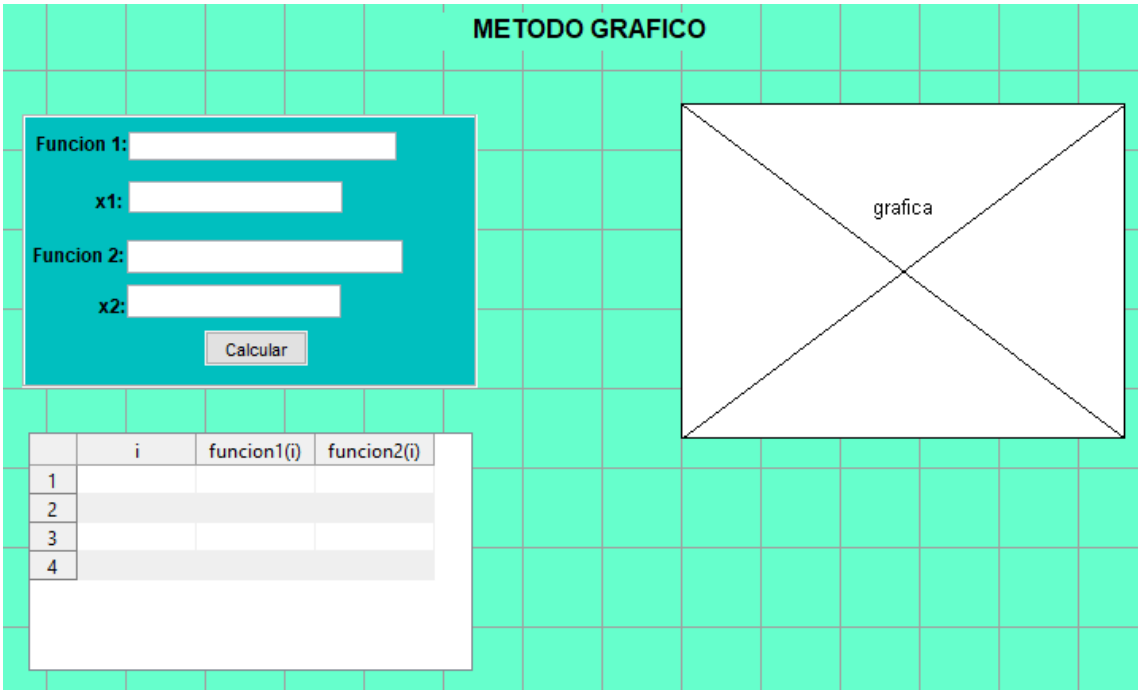
Eliminamos los textos por defecto.

Luego creamos un "***Table***", seguido doble clic y editamos en ***Tag*** "***tabla***".

Luego creamos un ***axes1*** para el grafico.

Finalmente creamo dos botones "***Calcular***" y "***Graficar***", con ***String*** "***calcular***" y ***tag*** "***btnCalcular***", luego el de limpiar seria ***String*** "***Limpiar***" con ***Tag*** "***btnLimpiar***".

Y quedaria de la siguiente manera:

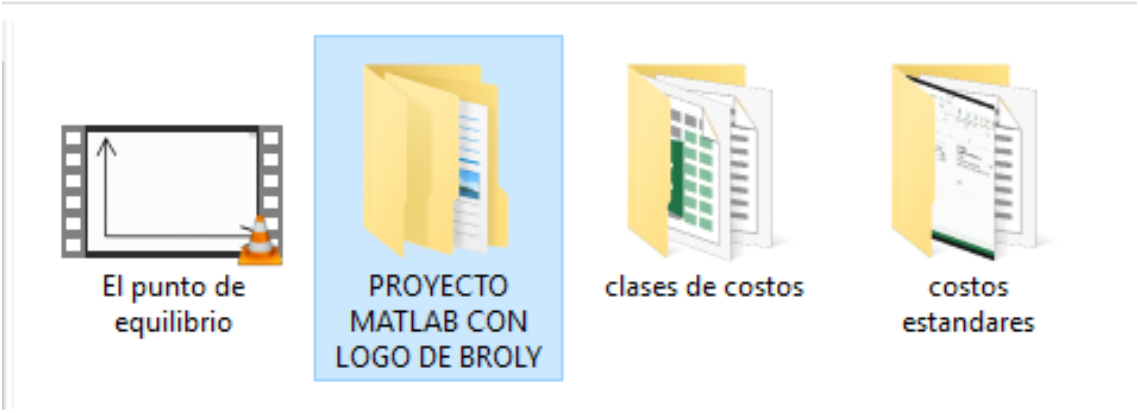


CODIFICACIÓN DEL BOTON CALCULAR

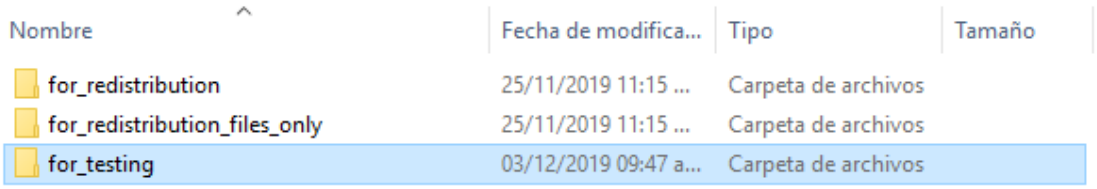
```
function btnCalcular_Callback(hObject, eventdata, handles)
funcion1=inline(get(handles.txtFuncion1,'string'));
x1 = str2num(get(handles.txtValor1,'string'));
funcion2 = inline(get(handles.txtFuncion2,'string'));
n= str2num(get(handles.txtNum,'string'))
set(handles.table,'Data',[]);
if isnan(x1)
    errordlg('Valor 1 Fuera de Rango','Error');
else

    for i=-(n+x1):1:n+x1
        newRow = {i,funcion1(i),funcion2(i)};
        oldData = get(handles.table,'Data');
        newData = [oldData; newRow];
        set(handles.table,'Data',newData)
        grid on;
        plot(i,funcion1(i),'*');
        grid on;
        plot(i,funcion2(i),'Ok')
        hold on;
    end
end
axes(handles.grafica);
grid
ezplot(funcion1);
grid
ezplot(funcion2);
hold on;
```

¿COMO UTILIZAR EL PROGRAMA Y EL APP?

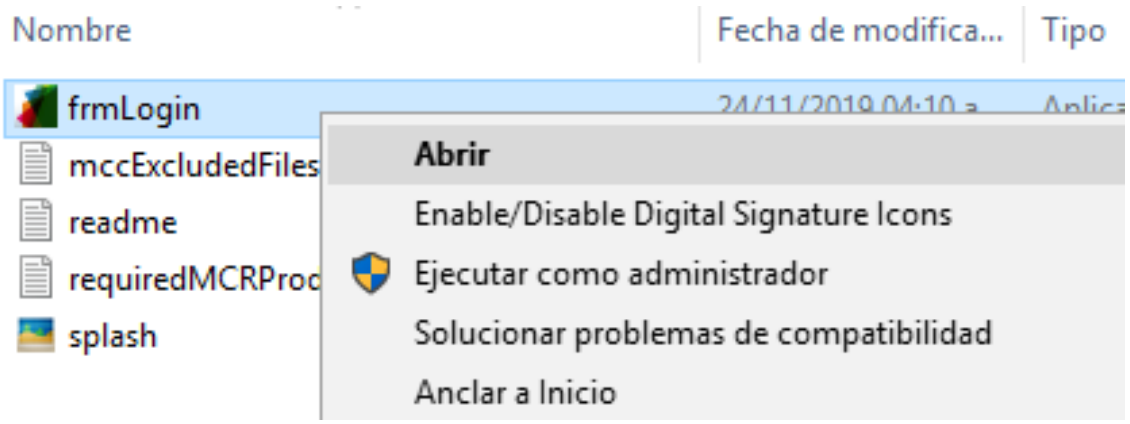
Primero ubicamos la carpeta en el pc o donde lo guardamos.



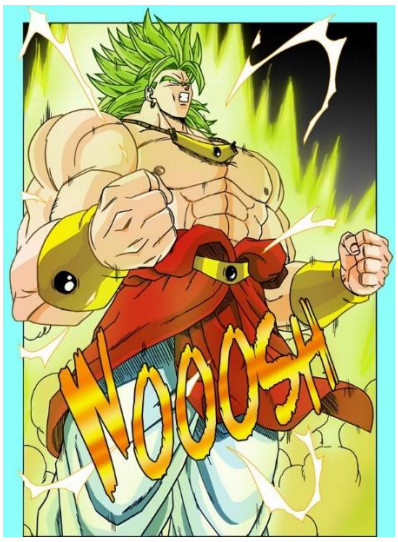
Abrimos la carpeta y abrimos la carpeta for-testing.



Luego procedemos a abrir la app

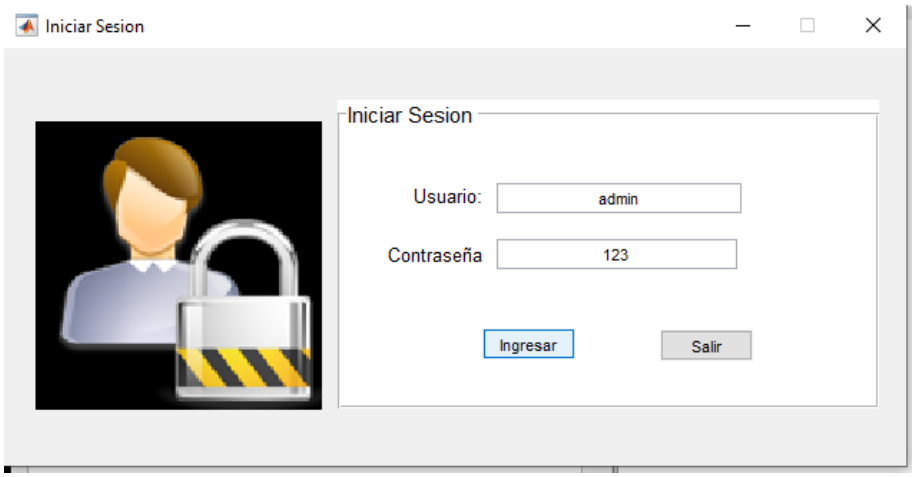


Abrimos la app esperar a que cargue te saldrá una imagen como esta.

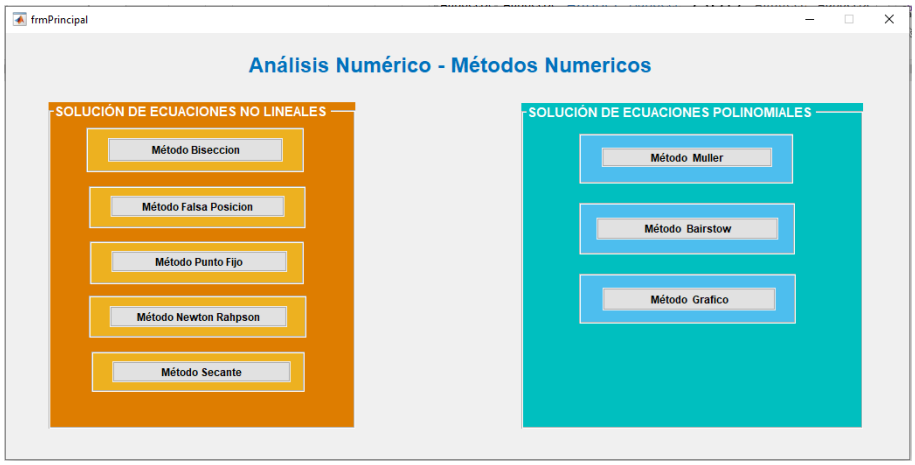


Usuario: admin

Contraseña: 123



Clic en ingresar y te saldrá la siguiente pantalla.



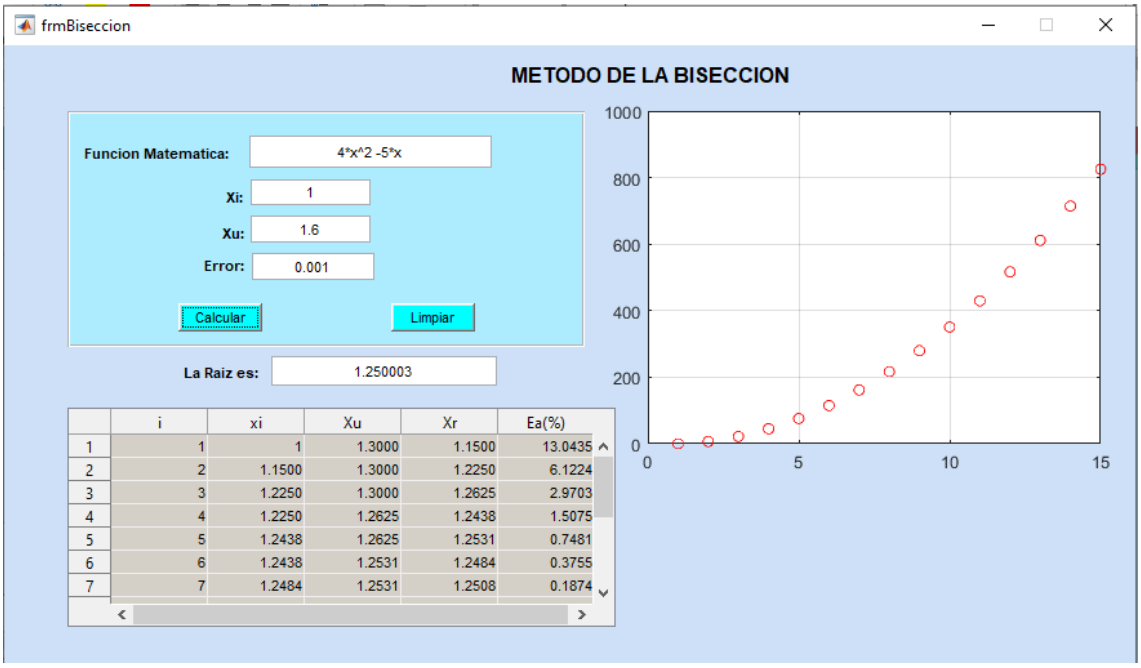
METODO DE BISECCIÓN

Clic en ***método de la bisección***.

En el programa de Matlab te saldrá la siguiente pantalla.

Y digitamos los datos.

Clic en ***calcular***.



Si deseas digitar otros datos clic en ***limpiar***.

Si deseas usar otro método salimos de esa pantalla dando clic en la parte superior derecha clic en la "x".

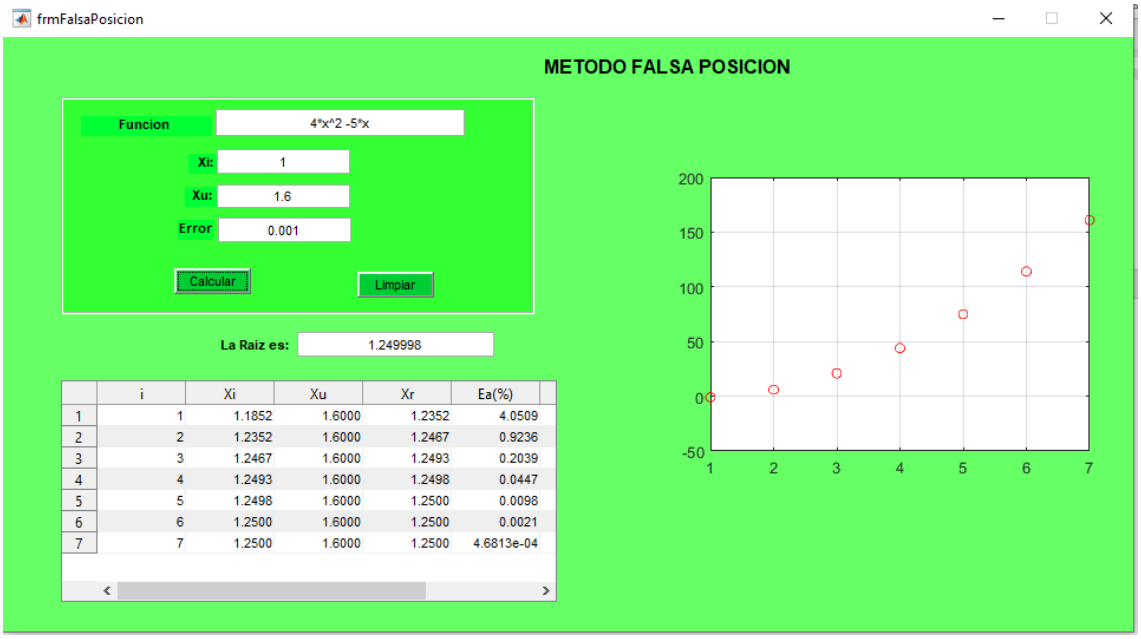
MÉTODO FALSA POSICIÓN

Luego clic en ***método de falsa posición***

Te saldrá la siguiente pantalla.

Digitamos los datos

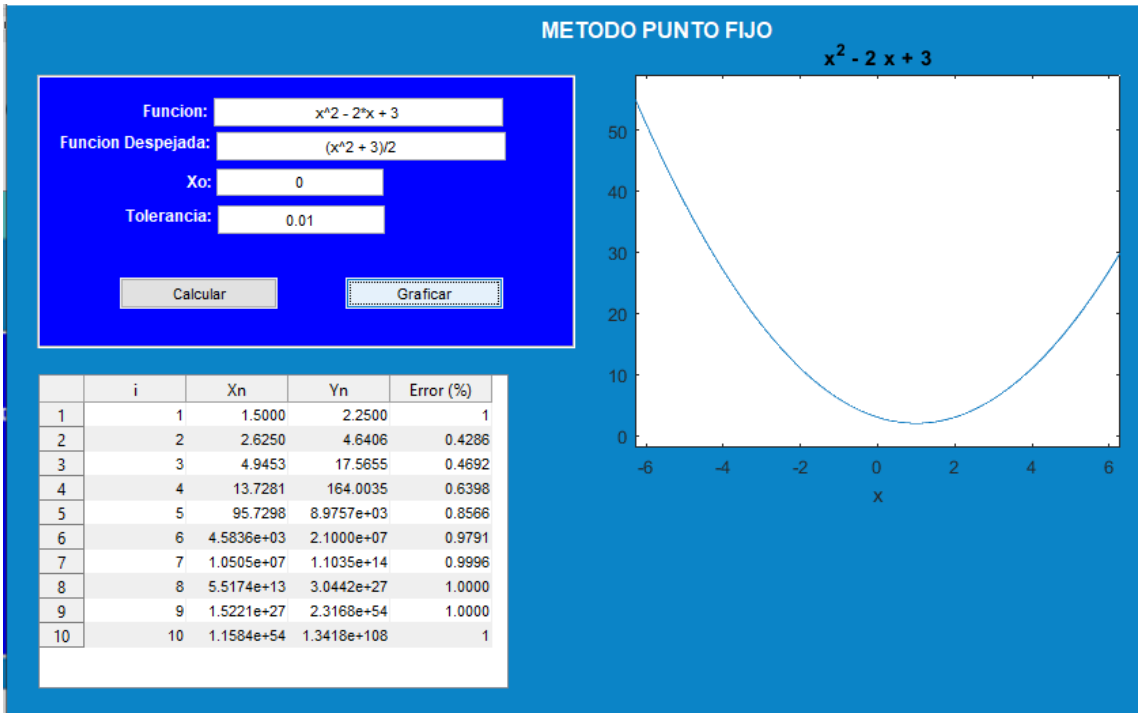
Clic en ***calcular***.



Si deseas introducir otros datos clic en ***limpiar***.

MÉTODO DEL PUNTO FIJO

Clic en ***método del punto fijo***.

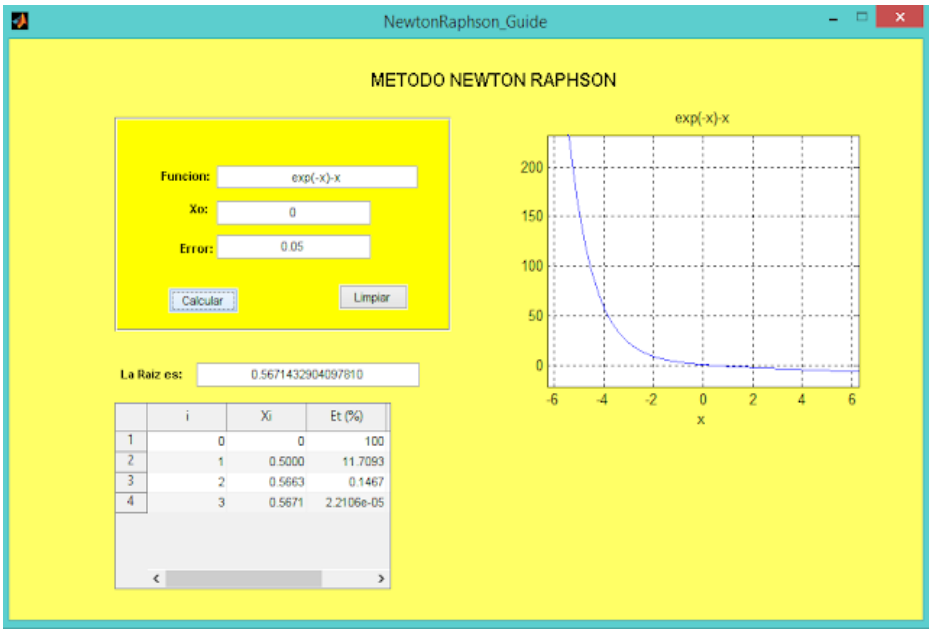


Digita la función y los otros datos luego clic en ***calcular*** y te saldrá el proceso si deseas la gráfica clic en ***gráfica*** y te mostrará la función gráficamente.

MÉTODO DE NEWTON-RAPHSON

Clic en ***método de newton-raphson***

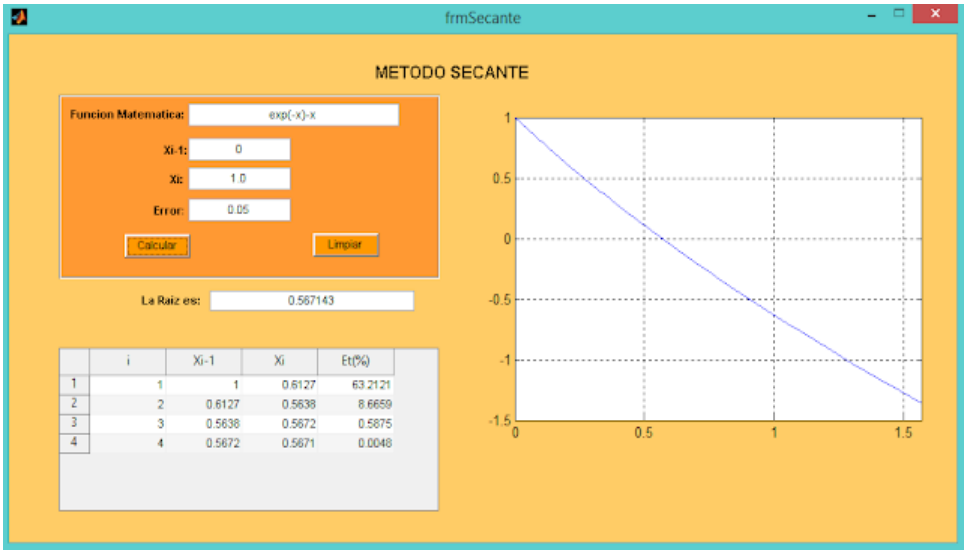
Introduces la función y los datos restantes clic en ***calcular*** y te mostrara los resultados en caso deseas introducir otra función clic en ***limpiar***.



MÉTODO DE LA SECANTE

Clic en ***método de la secante.***

Introduces la función y los datos restantes clic en ***calcular*** y te mostrara los resultados en caso deseas introducir otra función clic en ***limpiar***.



SOBRE LOS AUTORES

GUILLERMO AUGUSTO BOCANGEL WEYDERT: Destacado académico y líder en el ámbito de la Ingeniería Industrial y la Gestión del Conocimiento. Doctor en Ingeniería Industrial por la Universidad Nacional Federico Villareal, Magíster en Gestión del Conocimiento por la Escuela de Organización Industrial de España, Magister en Ingeniería Industrial en la Universidad Mayor de San Marcos, Post Doctor en Ciencias e Ingeniería y Doctor Honoris causa por la Universidad Nacional de Ucayali y la Universidad Intercultural de la Amazonia. Con una sólida trayectoria académica es docente en diversas universidades peruanas y extranjeras. Su expertise abarca la gerencia estratégica y los procesos, y es reconocido por su trabajo como consultor y conferencista internacional en temas de Balanced Scorecard (BSC) y control de procesos. https://orcid.org/0000-0003-1216-0944

ELMER S. CHUQUIYAURI SALDIVAR: Ingeniero de Sistemas e Informática, Magíster en Gestión Pública Para del Desarrollo Social, docente de la Universidad Nacional Hermilio Valdizán, Consultor en Modernización y Transformación Digital. Especialista en Desarrollo de Soluciones con TIC's. Universidad Nacional Hermilio Valdizán de Huánuco. https://orcid.org/0009-0009-4268-3034

GUILLERMO AUGUSTO BOCANGEL MARÍN: Ingeniero industrial con una destacada trayectoria académica y profesional, graduado de la Pontificia Universidad Católica del Perú (PUCP) con Maestría en Gestión y Dirección de Empresas Constructoras e Inmobiliarias y especialización en áreas como Automatización Industrial, Lean Manufacturing, Gestión de Proyectos y Sistemas de Gestión Integrados. Docente en la Facultad de Ingeniería y Arquitectura de la Universidad San Martín de Porres e investigador universitario. La combinación de su formación académica y su vasta experiencia empresarial le permite contribuir de manera significativamente al desarrollo de estrategias innovadoras y eficientes, tanto en el ámbito educativo como en el industrial. https://orcid.org/0000-0002-5431-9805

JHONNY HENRY PIÑÁN GARCÍA: Docente investigador. Ingeniero Industrial con especializaciones en Sistemas y Tecnologías de la Información, amplia experiencia en las áreas de desarrollo de sistemas y soporte tecnológico, en empresas líderes del país, con una Maestría en Didáctica y Tecnologías de la Información y egresado del Doctorado en Gestión Empresarial. Docente en la Universidad Nacional Hermilio Valdizán adscrito a la Facultad de Ingeniería Industrial. Autor de publicaciones de artículos científicos y de libros. Universidad Nacional Hermilio Valdizán de Huánuco. https://orcid.org/0000-0002-0263-7668

JORGE RUBÉN HILARIO CÁRDENAS: Ingeniero Industrial, Mg en Gestión y Planeamiento Educativo, Dr. en Gestión Empresarial, Posdoctor en Ciencias, Posdoctor en Investigación en Ingeniería e Innovación. Docente del pre y posgrado de la UNHEVAL, consultor experto en gestión empresarial, calidad, prospectiva, innovación e investigación científica, autor de libros y artículos científicos, Conferencista. https://orcid.org/0000-0001-6627-6489

NÉRIDA DEL CARMEN PASTRANA DÍAZ - Directora de transferencia e innovación en la UNHEVAL, autora de los libros “Aplicativos de Simulación de Sistemas Discretos en Empresas Productivas y de Servicios”, “Gestión de los Sistemas de Seguridad y Salud en el Trabajo” y Autora del artículo en la revista científica “Modelo de Medición del Capital Intelectual en las Carreras Acreditadas de Ingeniería Industrial del Perú”, ponente en eventos internacionales en temas sobre inteligencia de negocios y gestión del conocimiento, así mismo desarrollo trabajos de responsabilidad social sobre “Infraestructura Inalámbrica de Telecomunicaciones para Colegios de Zonas Rurales de la Región Huánuco. https://orcid.org/0000-0001-8357-3012

HERNAN WILMER GARCIA BONILLA: Ingeniero de sistemas, cuya formación y experiencia le ha permitido desarrollar un enfoque integral, especialmente en el análisis de sistemas y la gestión de información. Su pasión por entender la complejidad de las organizaciones complementa con un profundo interés en el análisis de datos y la prospectiva, capacitándolo para perfeccionar su capacidad para diseñar y gestionar sistemas que no solo respondan a las necesidades actuales, sino que también sean flexibles, escalables y adaptables a futuros desafíos. https://orcid.org/0009-0001-6542-079X

LINCOL JARLY GOMEZ MEZA: Ingeniero de sistemas con una sólida formación técnica y una Maestría en Gestión Pública, lo que me permite combinar experticia en desarrollo de software, redes y telecomunicaciones con una comprensión profunda de la administración pública y la gestión organizacional. Mi experiencia abarca una amplia gama de habilidades y conocimientos que me permiten abordar desafíos complejos desde una perspectiva integral. https://orcid.org/0000-0001-6173-5463