



TECNOLOGICO NACIONAL DE MEXICO

Instituto Tecnológico de la Laguna

Ingeniería en Sistemas Computacionales

TOPICOS AVANZADOS DE PROGRAMACION

PERIODO: Ene - Jun / 2020

GRUPO: "B" 17 – 18 Hrs

PRACTICA No. U1P04

EDITOR DE TEXTO BASICO CON JAVA SWING

ALUMNO:

18131209 José Misael Adame Sandoval

PROFESOR:

Ing. Luis Fernando Gil Vázquez

Torreón, Coah. a 14 de Marzo de 2020

Ejercicio 1

Desarrollar una aplicación con interfaz de usuario visual basada en Java Swing que implemente un Editor básico de archivos de texto plano. El Editor debe contar con las siguientes funcionalidades:

1. Crear un archivo nuevo
2. Abrir un archivo en disco
3. Guardar a un archivo en disco el texto del editor.
4. Imprimir el texto del editor
5. Proveer las funciones copiar, cortar, pegar para edición del texto.
6. Permitir cambiar el tipo y tamaño de letra.
7. Permitir cambiar el color de fondo y de la letra del texto del editor.
8. Proporcionar un minicorrector de palabras en el que tenga una lista predeterminada con palabras con errores comunes de escritura y su correspondiente corrección.
9. Proporcionar una función Acerca De que muestre el logo del tec e información completa de la institución, materia y datos de los autores.
10. La GUI debe proporcionar una barra de herramientas con botones de acceso rápido a las funciones. Los botones deben mostrar iconos intuitivos.
11. Al iniciar la ventana del editor debe mostrarse maximizada a todo el ancho y alto de la pantalla.
12. Al iniciar la aplicación debe mostrar una pantalla de bienvenida (splash) durante unos segundos.

Todos los controles visuales Java Swing que se incluyan en el código **deben tener un nombre de variable adecuado no el nombre de default**, ejemplo para un control JButton su variable asociada no se debe dejar en jButton1 se le debe poner un nombre tal como btnCalcular, etc. excepto para los JLabel o JPanel se pueden quedar como jLabel1, jLabel2, etc a menos que en código se modifique sus propiedades entonces deben llevar un nombre adecuado.

En la sección de ANALISIS presentar un “croquis” de la vista de la aplicación así como se esbozó en el pintarrón durante clases, donde indicamos los componentes visuales que integrarían la interfaz principal, los nombres de variables y describir la forma de abordar la implementación de las funciones de cambio de letra, colores y corrector.

En la sección de DISEÑO va el diagrama de clases UML de la aplicación.

En la sección de CODIGO va el código de todas las clases JAVA con la calidad establecida.

En la sección de PRUEBA DE EJECUCION pegar y describir las diversas pantallas de la aplicación.

Análisis

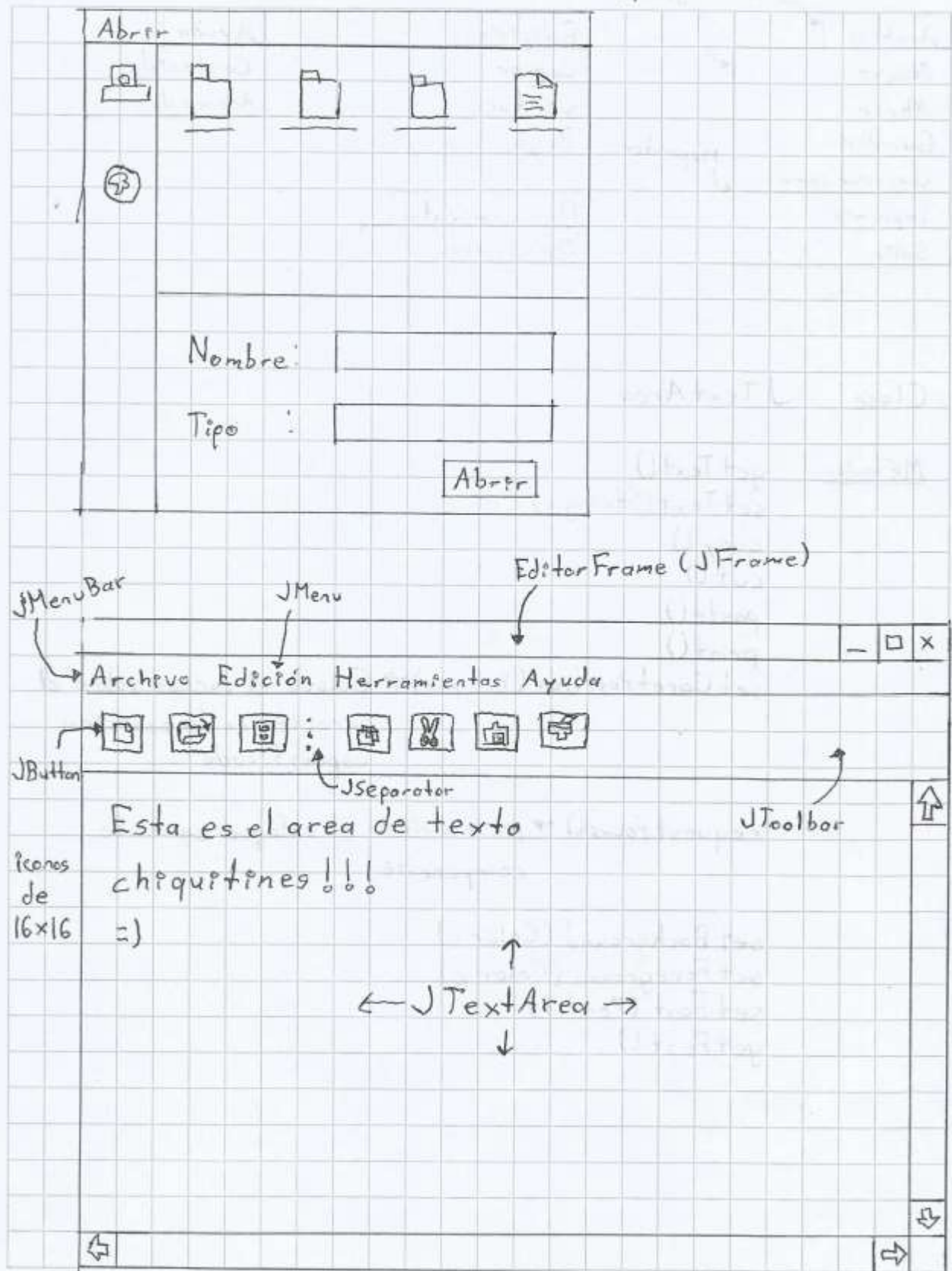
Basaremos el editor en un JTextArea que implementa funciones básicas de edición de texto plano, tales como: imprimir, copiar, pegar, cortar, manejar fuentes.

También se basará en manejo de archivos de texto (character streams) mediante clases tales como:

- File
- FileReader
- FileWriter
- BufferedReader

Aprovechamos la clase JFileChooser para mostrar diálogos estándar “Abrir” y “Guardar como”

Proyecto: UI Editor App





Clase: JTextArea

Métodos:

getText()

```
setText(String s)
```

copy()

$$cu^{\frac{1}{2}}()$$

paste()

```
print()
```

`setCaretPosition (int pos) →` Coloca el cursor sobre el caracter en la posición especificada

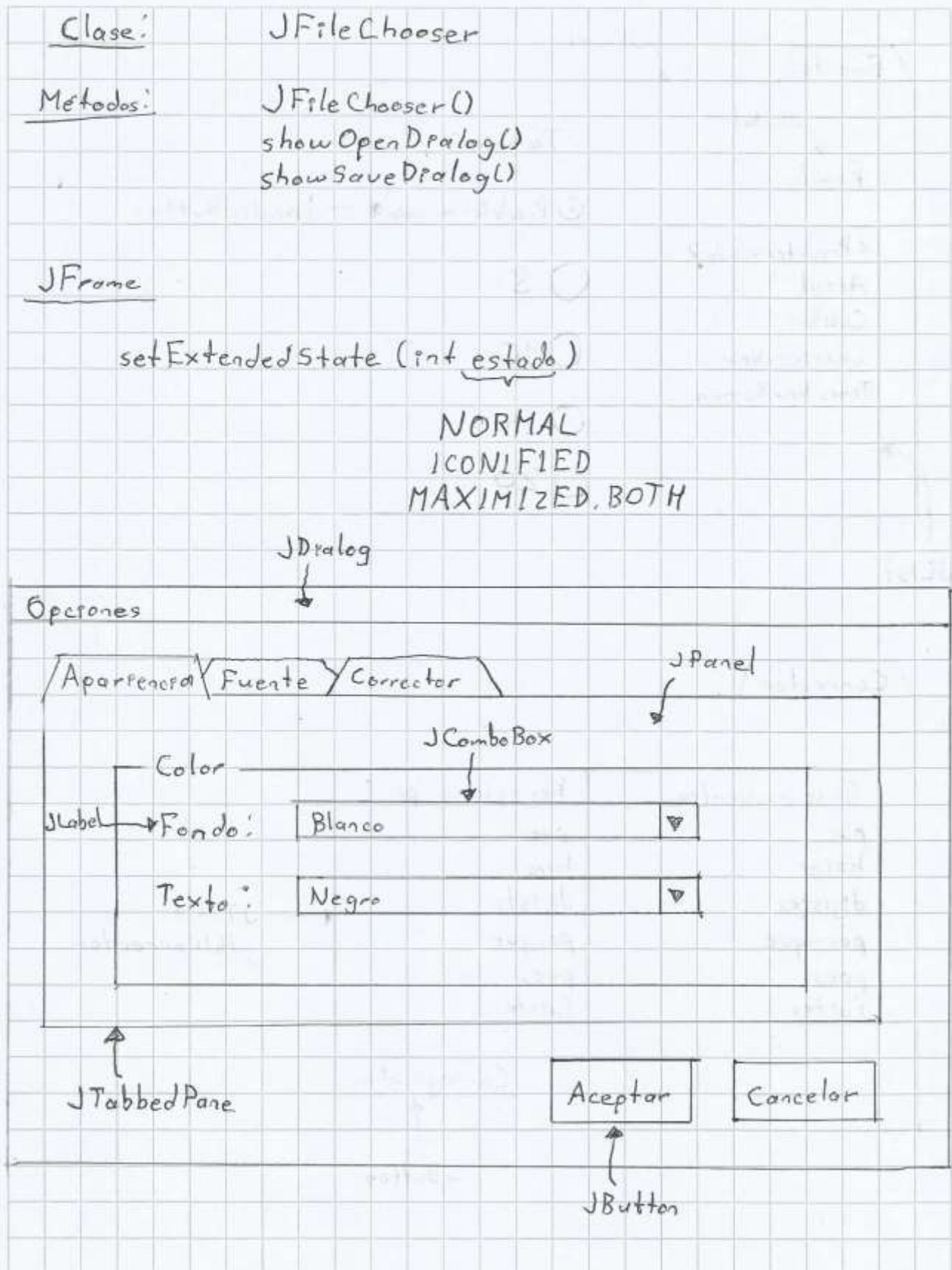
`requestFocus()` → para ubicar el enfoque en este componente

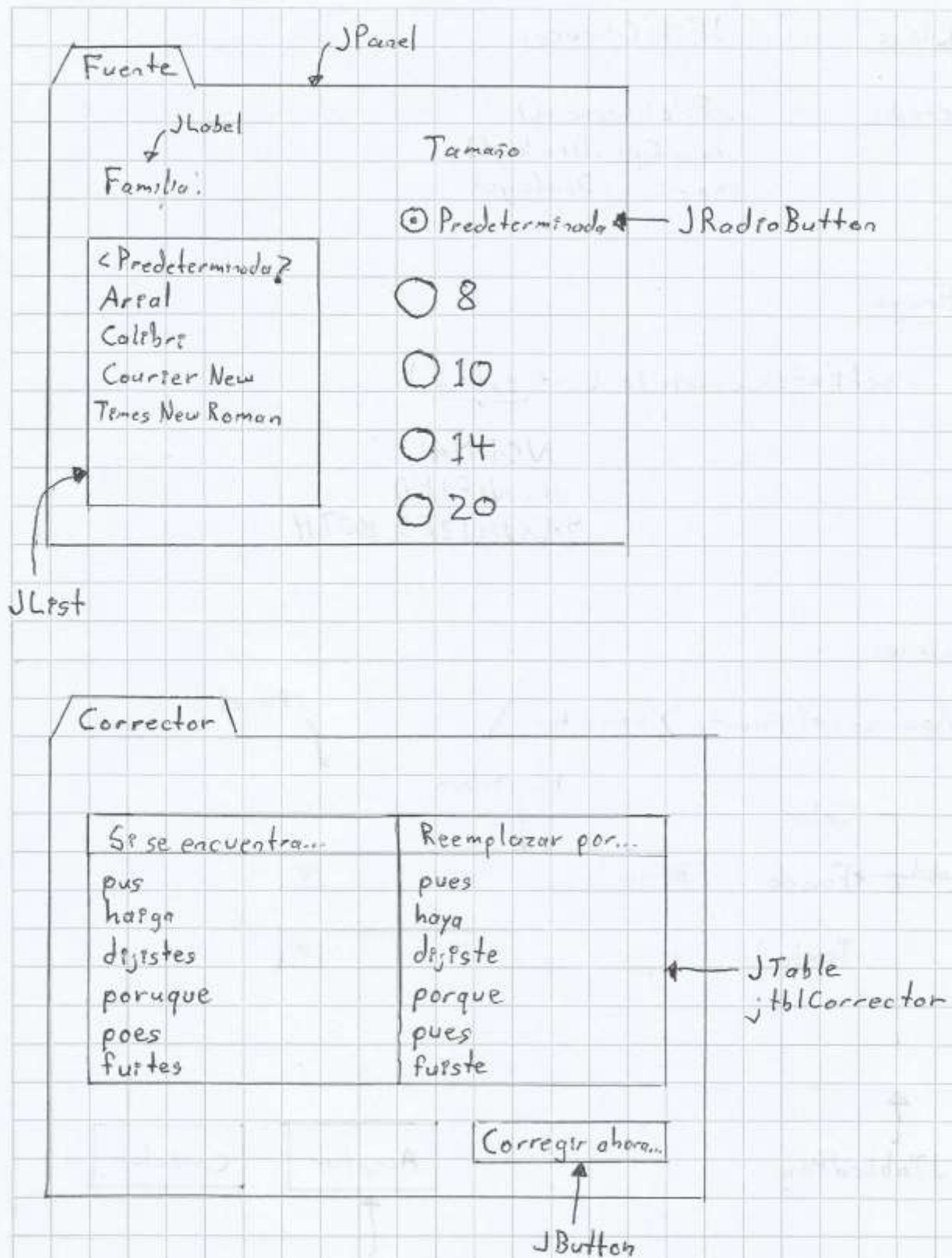
```
setBackground (Color c)
```

```
setForeground (Color c)
```

```
setFont(Font f)
```

getFont()





JTable

getRowCount()
getValueAt(int row, int col)

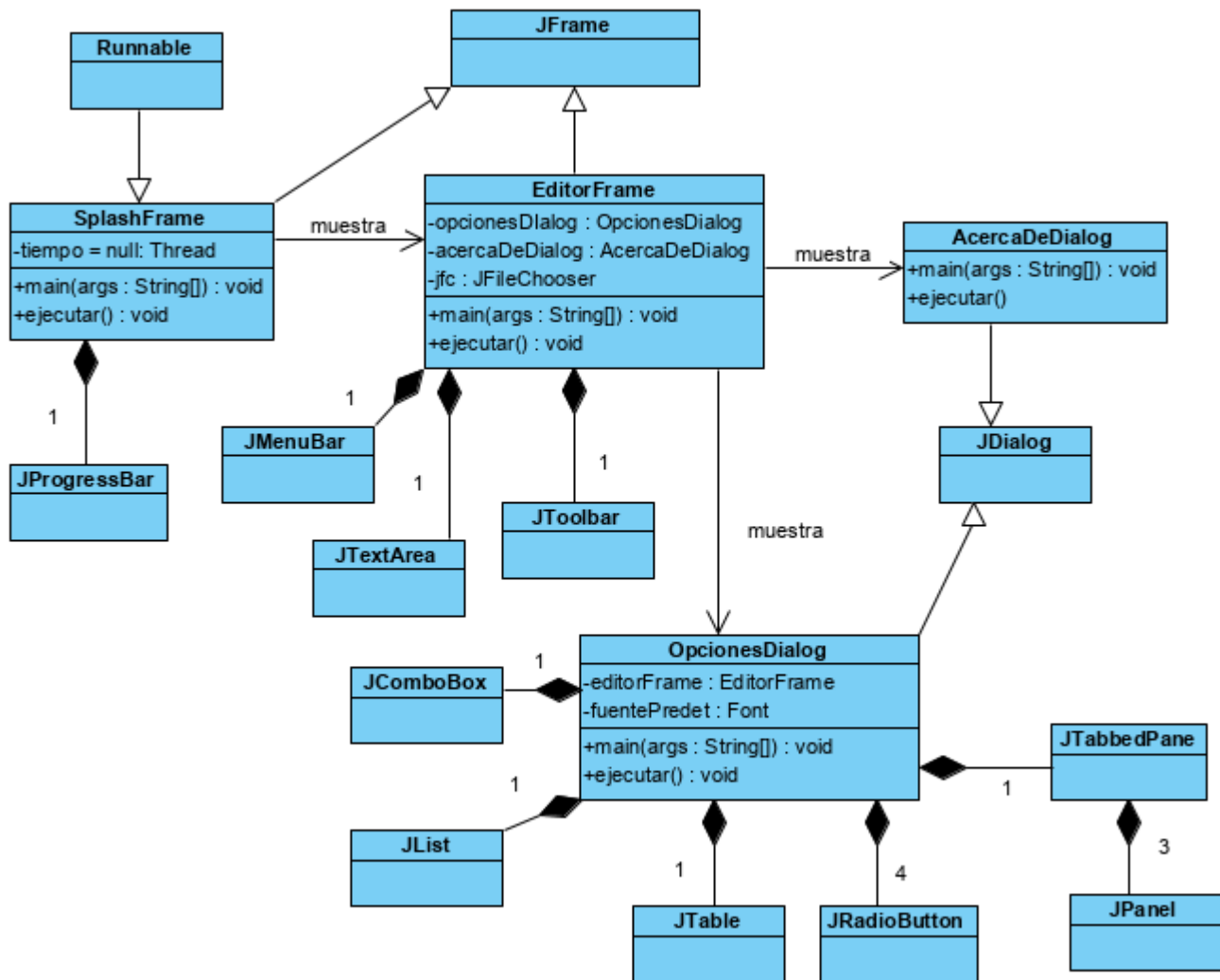
String

substring()
indexOf()

Abre un archivo pdf desde una aplicación Java

⇒ Clase: Desktop

Diseño



Código

AcercaDeDialog.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TÓPICOS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Aplicación GUI de un Acerca de..
*:
*: Archivo      : AcercaDeDialog.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 5/Mar/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Aplicación que muestra información sobre el autor de la
*:                la aplicación EditorApp
*:
*:
*: Última modif:
*: Fecha        Modificó          Motivo
*:-----
*: 6/Mar/2020 Misael Adame      Agregar prólogo.
*:-----*/
  
```



```

package editor;

import javax.swing.Icon;

public class AcercaDeDialog extends javax.swing.JDialog {

    //-----

    public AcercaDeDialog ( java.awt.Frame parent, boolean modal ) {
        super ( parent, true );
        initComponents();
        Icon ITL = Imagenes.escalarImagen ( jlblImagenITL.getIcon (),
                                            jlblImagenITL.getWidth (),
                                            jlblImagenITL.getHeight ());

        Icon TECNM = Imagenes.escalarImagen ( jlblImagenTECNM.getIcon (),
                                            jlblImagenTECNM.getWidth (),
                                            jlblImagenTECNM.getHeight ());

        jlblImagenITL.setIcon ( ITL );
        jlblImagenTECNM.setIcon ( TECNM );
    }

    //-----

    @SuppressWarnings("unchecked")
    Generated Code

    //-----

    private void jButtonCerrarActionPerformed(java.awt.event.ActionEvent evt) {
        dispose ();
    }

    //-----

    public static void main(String args[]) {
        ejecutar ();
    }

    //-----

    public static void ejecutar () {
        Look and feel setting code (optional)
        java.awt.EventQueue.invokeLater (new Runnable () {
            public void run () {
                AcercaDeDialog dialog = new AcercaDeDialog ( new javax.swing.JFrame (), true);
                dialog.addWindowListener ( new java.awt.event.WindowAdapter () {
                    @Override
                    public void windowClosing( java.awt.event.WindowEvent e ) {
                        System.exit ( 0 );
                    }
                });
                dialog.setVisible ( true );
            }
        });
    }
}

```

Imagenes.java

```

/*-----
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Clase con método que permite ajustar el ancho y alto de una imagen
*:
*:
*: Archivo      : Imagenes.java

```

```

:* Autor      : Ing. Fernando Gil
:* Fecha      : 29/08/2018
:* Compilador : JAVA J2SE v1.8.0
:* Descripción : El metodo estatico escalarImagen () es el más adecuado de usar para
:*              ajustar el ancho y alto de la imagen de un objeto Icon.
:*              El metodo recibe 3 argumentos: el objeto Icon y el ancho y alto al que
:*              se desea ajustar. El metodo devuelve un objeto Icon con la imagen ya
:*              redimensionada.
:*
:* Ultima modif:
:* Fecha      Modific?          Motivo
:*=====
:* 18/02/2020 F.Gil             -Corregir informacion del prologo.
:*-----*/

```

```
package prismas;
```

```

import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.RenderingHints;
import java.awt.image.BufferedImage;
import javax.swing.Icon;
import javax.swing.ImageIcon;

public class Imagenes {

    //-----

    public static Image getScaledImage ( Image srcImg, int w, int h ) {
        BufferedImage resizedImg = new BufferedImage(w, h, BufferedImage.TYPE_INT_RGB);
        Graphics2D g2 = resizedImg.createGraphics();
        g2.setRenderingHint(RenderingHints.KEY_INTERPOLATION, RenderingHints.VALUE_INTERPOLATION_BILINEAR);
        g2.drawImage(srcImg, 0, 0, w, h, null);
        g2.dispose();
        return resizedImg;
    }

    //-----

    public static Icon escalarImagen ( Icon srcImg, int w, int h ) {
        Image img = ( (ImageIcon) srcImg ).getImage ();
        img = getScaledImage ( img, w, h );
        return new ImageIcon ( img );
    }

    //-----
}

```

EditorFrame.java

```

*-----
:*              INSTITUTO TECNOLÓGICO DE LA LAGUNA
:*              INGENIERÍA EN SISTEMAS COMPUTACIONALES
:*              TEMAS AVANZADOS DE PROGRAMACIÓN "B"
:*
:*              SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
:*
:*              Aplicación GUI para el entorno del Editor de Texto
:*
:* Archivo      : EditorFrame.java
:* Autor        : José Misael Adame Sandoval      18131209
:* Fecha        : 04/Mar/2020
:* Compilador   : JAVA J2SE v1.8.0
:* Descripción  : Aplicación que realiza las funciones básicas de un editor
:*              de texto básico como crear un archivo, abrirlo, guardarlo,
:*              imprimirlo, copiar, cortar y pegar el texto.
:*
:*
:* Ultima modif:
:* Fecha        Modificó          Motivo
:*=====
:* 6/mar/2020 Misael Adame        Agregar prólogo.

```

```
/*-----*/

package editor;

import java.awt.Desktop;
import java.awt.print.PrinterException;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;

public class EditorFrame extends javax.swing.JFrame {

    //-----

    private OpcionesDialog opcionesDialog;
    private AcercaDeDialog acercaDeDialog;
    private JFileChooser jfc;

    //-----

    public EditorFrame () {
        initComponents ();

        opcionesDialog = new OpcionesDialog ( this, true );
        jfc = new JFileChooser ();

        // Maximizar el frame
        this.setExtendedState ( MAXIMIZED_BOTH );
    }

    //-----
    @SuppressWarnings("unchecked")
    Generated Code
    //-----

    private void jMenuItemArchivoNuevoActionPerformed(java.awt.event.ActionEvent evt) {
        if ( JOptionPane.showConfirmDialog (
            this,
            "¿Desea crear un archivo nuevo?",
            "Editor",
            JOptionPane.YES_NO_OPTION,
            JOptionPane.QUESTION_MESSAGE ) == JOptionPane.YES_OPTION ) {
            jtxaTexto.setText ( "" );
            jtxaTexto.requestFocus ();
        }
    }

    //-----

    private void jMenuItemArchivoAbrirActionPerformed(java.awt.event.ActionEvent evt) {
        if ( jfc.showOpenDialog ( this ) == JFileChooser.APPROVE_OPTION ) {
            File file = jfc.getSelectedFile ();
            try {
                FileReader fr = new FileReader ( file );
                BufferedReader br = new BufferedReader ( fr );
                String texto = "";
                String linea = br.readLine ();
                while ( linea != null ) {
                    texto += linea + "\n";
                    linea = br.readLine ();
                }
                br.close ();
                fr.close ();
                jtxaTexto.setText ( texto );
                jtxaTexto.setCaretPosition ( 0 );
                this.setTitle ("EditorApp - " + jfc.getName ( jfc.getSelectedFile() ) );
            } catch ( FileNotFoundException ex ) {
                JOptionPane.showMessageDialog ( this, ex, "Error", JOptionPane.ERROR_MESSAGE );
            } catch ( IOException ex ) {

```

```
        JOptionPane.showMessageDialog ( this, ex, "Error", JOptionPane.ERROR_MESSAGE );
    }
}

//-----

private void jmniArchivoGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        FileWriter fw = new FileWriter ( jfc.getSelectedFile () );
        String texto = jtxaTexto.getText ();
        fw.write ( texto );
        fw.close ();
    } catch ( IOException ex ) {
        JOptionPane.showMessageDialog ( this, ex, "Error", JOptionPane.ERROR_MESSAGE );
    } catch ( NullPointerException e ) {
        if ( jfc.showSaveDialog ( this ) == JFileChooser.APPROVE_OPTION ) {
            try {
                FileWriter fw = new FileWriter ( jfc.getSelectedFile () );
                String texto = jtxaTexto.getText ();
                fw.write ( texto );
                fw.close ();
                this.setTitle ("EditorApp - " + jfc.getName( jfc.getSelectedFile() ) );
            } catch ( IOException ex ) {
                JOptionPane.showMessageDialog ( this, ex, "Error", JOptionPane.ERROR_MESSAGE );
            }
        }
    }
}

//-----

private void jmniArchivoSalirActionPerformed(java.awt.event.ActionEvent evt) {
    dispose ();
}

//-----

private void jbtnArchivoNuevoActionPerformed(java.awt.event.ActionEvent evt) {
    jmniArchivoNuevoActionPerformed ( evt );
    // jmniArchivoNuevo.doClick ();
}

//-----

private void jmniArchivoImprimirActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        jtxaTexto.print ();
    } catch ( PrinterException ex ) {
        JOptionPane.showMessageDialog(
            this,
            ex,
            "Error",
            JOptionPane.ERROR_MESSAGE
        );
    }
}

//-----

private void jmniEditorCortarActionPerformed(java.awt.event.ActionEvent evt) {
    jtxaTexto.cut ();
}

//-----

private void jmniEditorCopiarActionPerformed(java.awt.event.ActionEvent evt) {
    jtxaTexto.copy ();
}

//-----

private void jmniHerramientasOpcionesActionPerformed(java.awt.event.ActionEvent evt) {
    opcionesDialog.setVisible ( true );
}
```

```
private void jmnIAyudaAcercaDeActionPerformed(java.awt.event.ActionEvent evt) {
    acercaDeDialog = new AcercaDeDialog ( this, true );
    acercaDeDialog.setVisible ( true );
}

//-----

private void jmnIArchivoGuardarComoActionPerformed(java.awt.event.ActionEvent evt) {
    jfc.setDialogTitle ( "Guardar como..." );
    if ( jfc.showSaveDialog ( this ) == JFileChooser.APPROVE_OPTION ) {
        try {
            FileWriter fw = new FileWriter ( jfc.getSelectedFile() );
            String texto = jtxaTexto.getText ();
            fw.write ( texto );
            fw.close ();
            this.setTitle ("EditorApp - " + jfc.getName( jfc.getSelectedFile() ) );
        } catch ( IOException ex ) {
            JOptionPane.showMessageDialog ( this, ex, "Error", JOptionPane.ERROR_MESSAGE );
        }
    }
}

//-----

private void jbtnEditarPegarActionPerformed(java.awt.event.ActionEvent evt) {
    jmnIEditarPegarActionPerformed ( evt );
}

//-----

private void jbtnArchivoAbrirActionPerformed(java.awt.event.ActionEvent evt) {
    jmnIArchivoAbrirActionPerformed ( evt );
}

//-----

private void jbtnArchivoGuardarActionPerformed(java.awt.event.ActionEvent evt) {
    jmnIArchivoGuardarActionPerformed ( evt );
}

//-----

private void jbtnEditarCopiarActionPerformed(java.awt.event.ActionEvent evt) {
    jmnIEditarCopiarActionPerformed ( evt );
}

//-----

private void jbtnEditarCortarActionPerformed(java.awt.event.ActionEvent evt) {
    jmnIEditarCortarActionPerformed ( evt );
}

//-----

private void jmnIEditarPegarActionPerformed(java.awt.event.ActionEvent evt) {
    jtxaTexto.paste ();
}

//-----

private void jmnIAyudaContenidoActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Desktop.getDesktop ().open ( new java.io.File ( "TAP-U1P04-Editor de texto básico con "
            + "Java Swing.pdf" ) );
    } catch ( IOException ex ) {
        JOptionPane.showMessageDialog (
            this,
            "Archivo no encontrado",
            "Error",
            JOptionPane.ERROR_MESSAGE
        );
    }
}
```



```

//-----
public static void ejecutar () {
    Look and feel setting code (optional)

    java.awt.EventQueue.invokeLater ( new Runnable () {
        public void run () {
            new EditorFrame ().setVisible ( true );
        }
    });
}

//-----

public static void main ( String args[] ) {
    ejecutar ();
}

//-----
}

```

OpcionesDialog.java

```

/*-----
*:
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Aplicación GUI para las Opciones del Editor de Texto
*:
*: Archivo      : OpcionesDialog.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 04/Mar/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Aplicación que realiza modificaciones en el editor
*:                  1. Apariencia: permite modificar el color de la fuente y el fondo
*:                  del editor
*:                  2. Fuente: permite modificar el estilo de la fuente y el tamaño
*:                  3. Corrector: Si encuentra unas palabras determinadas
*:                  mal escritas las corrige.
*:
*:
*: Última modif:
*: Fecha      Modificó      Motivo
*:=====
*: 6/mar/2020 Misael Adame      Agregar prólogo.
*:-----*/

package editor;

import java.awt.Color;
import java.awt.Font;
import javax.swing.JOptionPane;

public class OpcionesDialog extends javax.swing.JDialog {

    //-----

    private EditorFrame editorFrame;
    private Font fuentePredet;

    //-----

    public OpcionesDialog ( java.awt.Frame parent, boolean modal ) {
        super ( parent, modal );
        initComponents ();

        // Conservar la referencia a EditorFrame
        editorFrame = ( EditorFrame ) parent;
        // Guardamos la fuente predeterminada del TextArea
        fuentePredet = editorFrame.jtxaTexto.getFont ();
    }
}

```

```

}

//-----

@SuppressWarnings("unchecked")
Generated Code

//-----

private void jButtonAceptarActionPerformed(java.awt.event.ActionEvent evt) {
    // Cambiar el color del fondo y del texto
    switch ( jcbColorFondo.getSelectedIndex () ) {
        case 0 : editorFrame.jtxaTexto.setBackground ( Color.white ); break;
        case 1 : editorFrame.jtxaTexto.setBackground ( Color.black ); break;
        case 2 : editorFrame.jtxaTexto.setBackground ( Color.blue ); break;
        case 3 : editorFrame.jtxaTexto.setBackground ( Color.red ); break;
    }

    switch ( jcbColorTexto.getSelectedIndex () ) {
        case 0 : editorFrame.jtxaTexto.setForeground ( Color.white ); setVisible ( false ); break;
        case 1 : editorFrame.jtxaTexto.setForeground ( Color.black ); setVisible ( false ); break;
        case 2 : editorFrame.jtxaTexto.setForeground ( Color.green ); setVisible ( false ); break;
        case 3 : editorFrame.jtxaTexto.setForeground ( Color.blue ); setVisible ( false ); break;
    }

    // Cambiar la fuente y Tamaño
    String familia = jlstFamilia.getSelectedValue ();
    if ( familia.equals ( "<Predeterminada>" ) )
        familia = fuentePredet.getFamily ();
    int tamano = fuentePredet.getSize ();
    if ( jrbtTam8.isSelected () )
        tamano = 8;
    else if ( jrbtTam10.isSelected () )
        tamano = 10;
    else if ( jrbtTam14.isSelected () )
        tamano = 14;
    else if ( jrbtTam20.isSelected () )
        tamano = 20;
    Font f = new Font ( familia, Font.PLAIN, tamano );
    editorFrame.jtxaTexto.setFont ( f );
}

//-----

private void jButtonCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    setVisible ( false );
}

//-----

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String texto = editorFrame.jtxaTexto.getText();
    int totFilas = jTableCorrector.getRowCount ();
    for ( int i = 0; i < totFilas; i++ ) {
        // Obtenemos de la tabla corrector la palabra en turno a procesar
        String buscar = jTableCorrector.getValueAt ( i, 0 ).toString();
        String reemplazarPor = jTableCorrector.getValueAt ( i, 1 ).toString();

        // Proceder a la búsqueda y reemplazo
        int pos = texto.indexOf ( buscar );
        while ( pos != -1 ) {
            texto = texto.substring(0, pos) + reemplazarPor +
                texto.substring(pos + buscar.length());
            pos = texto.indexOf ( buscar, pos + 1);
        }
        // Sustituir el texto del editor con el texto corregido
        editorFrame.jtxaTexto.setText ( texto );
    }
    JOptionPane.showMessageDialog (
        this,
        "Corrección finalizada!",
        "Corrector",
        JOptionPane.INFORMATION_MESSAGE
    );
}

```

```

    );
}

//-----

public static void main(String args[]) {
    ejecutar ();
}

//-----

public static void ejecutar () {

    Look and feel setting code (optional)

    /* Create and display the dialog */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            OpcionesDialog dialog = new OpcionesDialog(new javax.swing.JFrame(), true);
            dialog.addWindowListener(new java.awt.event.WindowAdapter() {
                @Override
                public void windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}
}

```

SplashFrame.java

```

/*-----
*:          INSTITUTO TECNOLÓGICO DE LA LAGUNA
*:          INGENIERÍA EN SISTEMAS COMPUTACIONALES
*:          TEMAS AVANZADOS DE PROGRAMACIÓN "B"
*:
*:          SEMESTRE: ENE-JUN/2020      HORA: 17-18 HRS
*:
*:          Aplicación GUI Splash
*:
*: Archivo      : SplashFrame.java
*: Autor        : José Misael Adame Sandoval      18131209
*: Fecha        : 04/Mar/2020
*: Compilador   : JAVA J2SE v1.8.0
*: Descripción  : Aplicación que hace de Splash (antes de iniciar el
*:                EditorFrame.java)
*:
*: Última modif:
*: Fecha        Modificó          Motivo
*:-----
*: 06/mar/2020 Misael Adame      Agregar prólogo.
*:-----*/

package editor;

import com.sun.awt.AWTUtilities;
import java.util.logging.Level;
import java.util.logging.Logger;

public class SplashFrame extends javax.swing.JFrame implements Runnable {

    //-----

    private Thread tiempo = null;

    //-----

    public SplashFrame() {
        initComponents();
        this.setLocationRelativeTo(null);
    }
}

```

```

        AWTUtilities.setWindowOpaque(this, false);
        tiempo = new Thread ( this );
        tiempo.start();
    }

    //-----

    @SuppressWarnings("unchecked")
    Generated Code

    //-----

    public static void main(String args[]) {
        ejecutar();
    }

    //-----

    public static void ejecutar() {
        Look and feel setting code (optional)
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new SplashFrame().setVisible(true);
            }
        });
    }

    //-----

    @Override
    public void run() {
        while ( tiempo != null){
            try {
                for ( int i = 0; i <= 100; i++) {
                    Thread.sleep(37);
                    this.jlblPorcBarraProgreso.setText(Integer.toString(i)+" %");
                    this.jProgressBarSplash.setValue(i);
                    if( i == 100 ) {
                        this.setVisible(false);
                        Thread.sleep(600);
                        new EditorFrame().setVisible(true);
                    }
                }
                break;
            } catch (InterruptedException ex) {
                Logger.getLogger(SplashFrame.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        tiempo = null;
    }
}

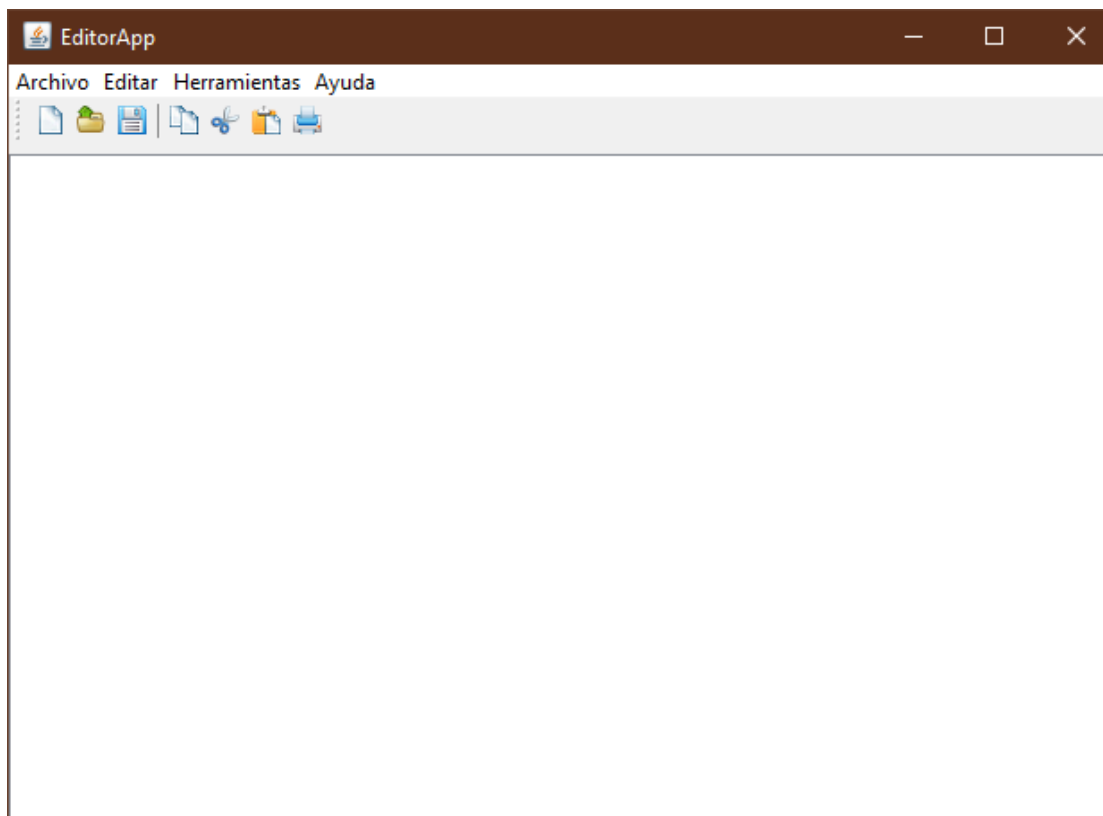
```

Prueba de Ejecución

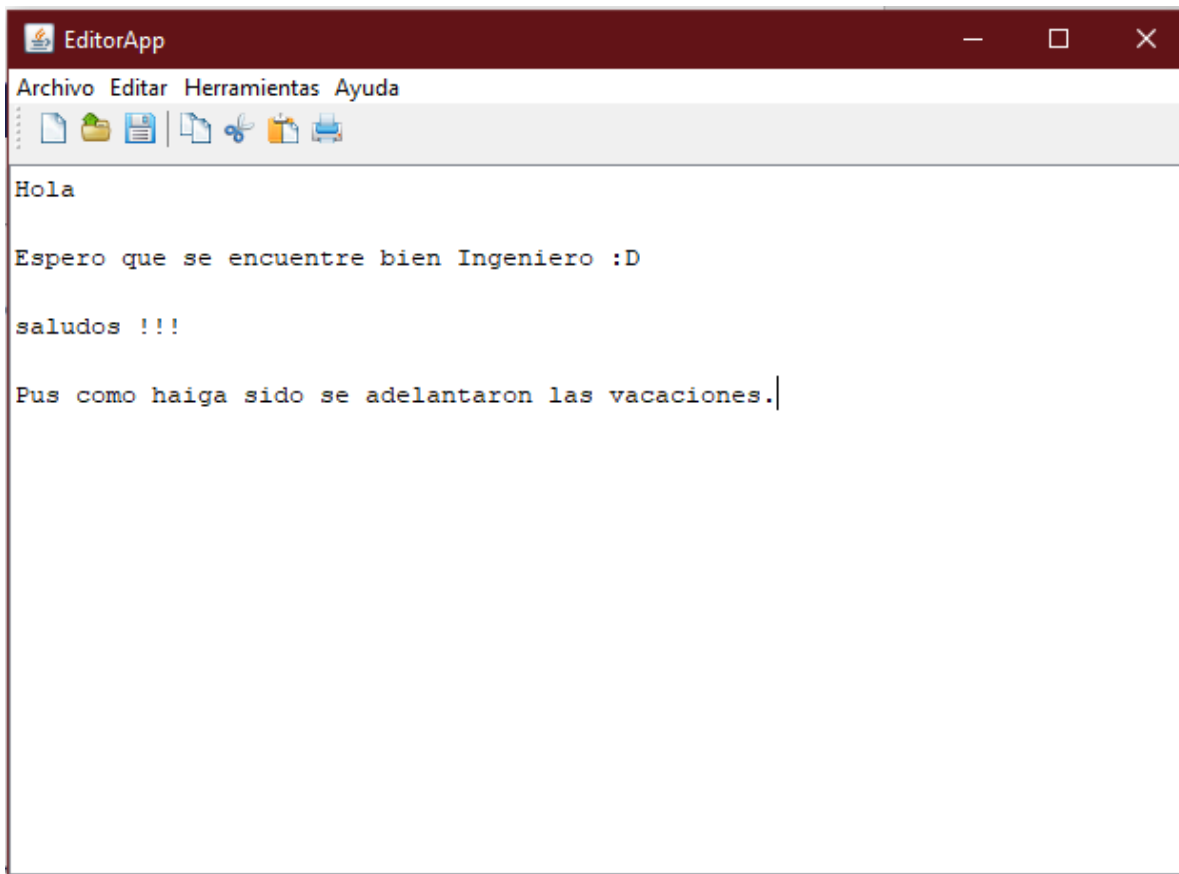
Este es el Splash al ejecutar la aplicación



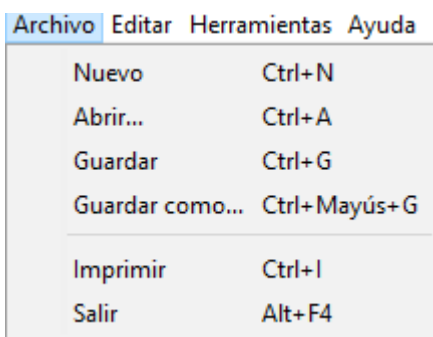
Esta es la interfaz inicial



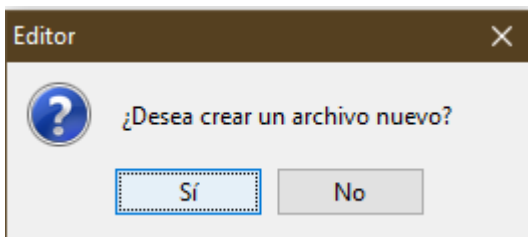
Escribir algo en el editor de texto



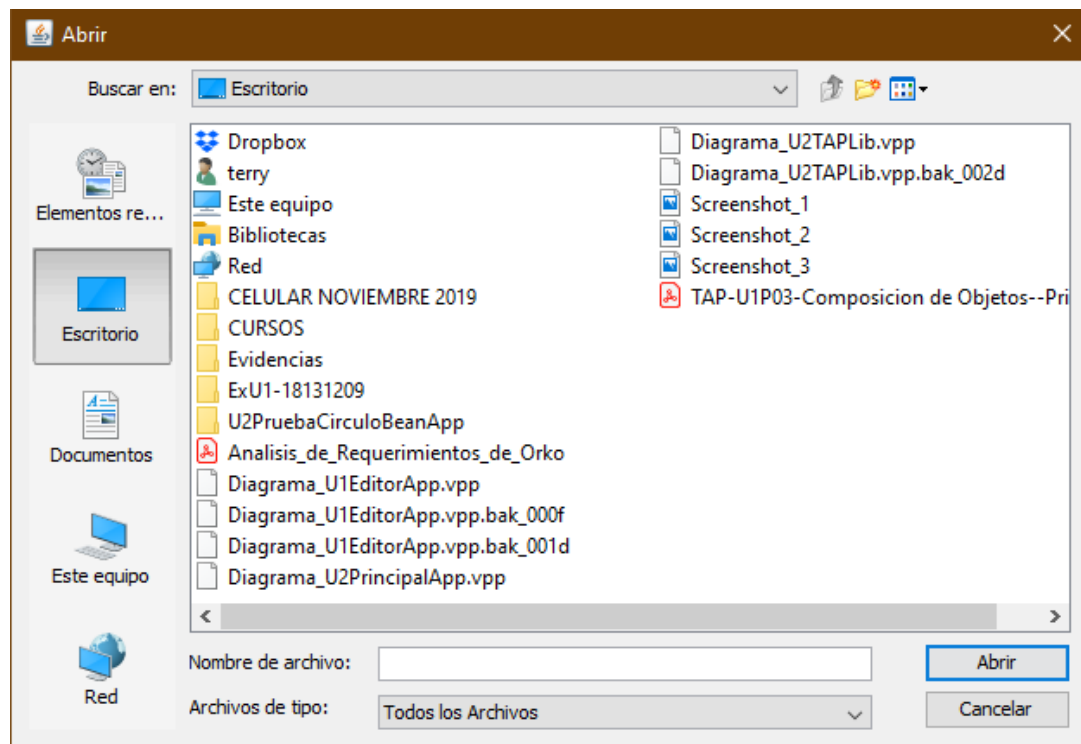
Al abrir las opciones de Archivo se despliega



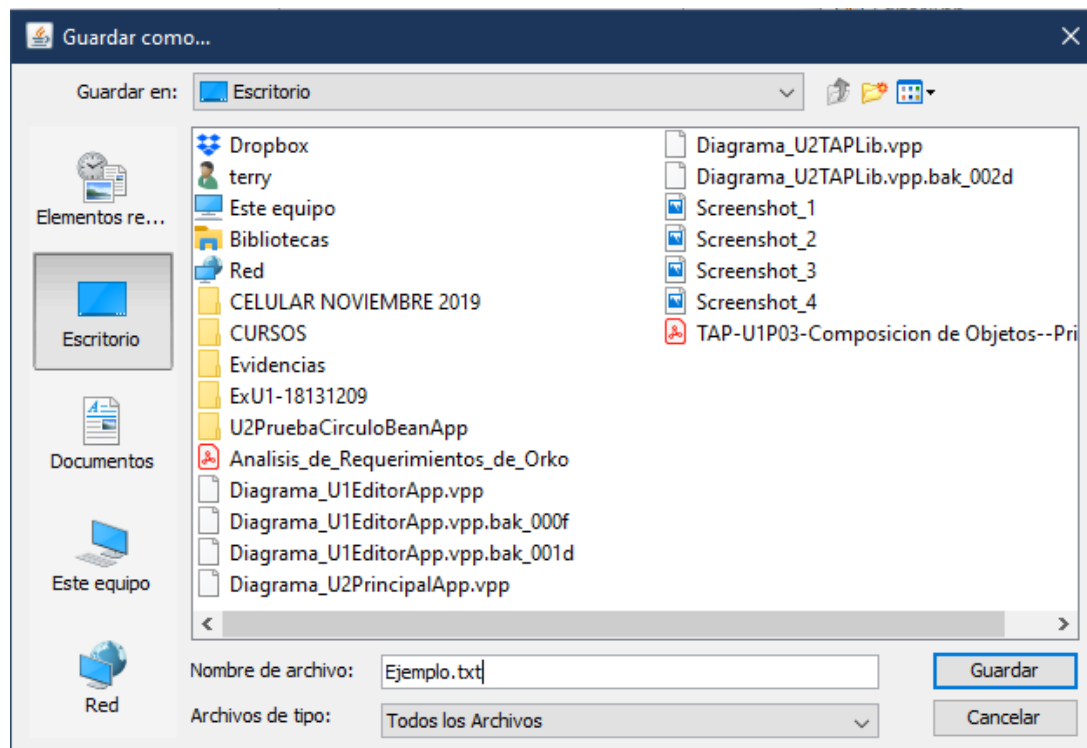
Al dar clic en Nuevo, se despliega un cuadro de dialogo en el que si da clic en "Sí" se crea un nuevo archivo sin texto, si le dio clic en "No" no hace nada.



Al dar clic en Abrir..., se despliega un diálogo para escoger el archivo de texto que desea abrir, y al dar clic en Abrir al haber seleccionado su archivo, se despliega la información del archivo en el editor de texto, si le dio clic en Cancelar no realiza nada



Al realizar clic en Guardar se guarda los cambios que realizo al archivo en el editor de texto, pero en caso de que no exista el archivo todavía realiza la función de Guardar Como... en la que abre un dialogo de guardar (estéticamente similar al Abrir...), en la que el usuario escoge la ubicación y nombre en la que guardará el archivo (en este caso se llama Ejemplo.txt y se guardará en el Escritorio), al dar clic en Guardar realiza lo anteriormente mencionado en caso de dar clic en Cancelar no realiza nada.



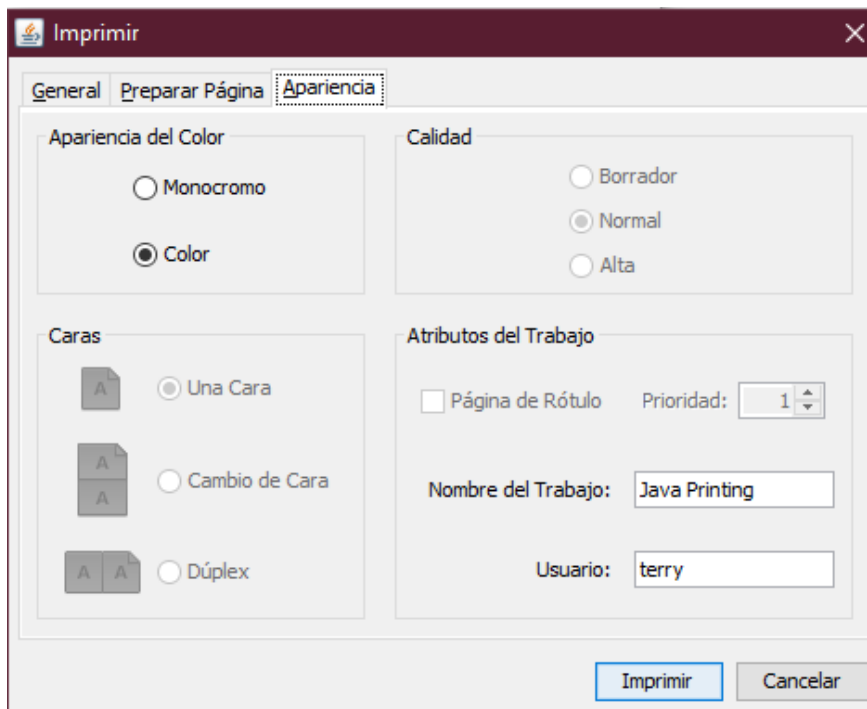
Al dar clic en Imprimir se abre una ventana de diálogo en la que puedes hacer ajustes en la impresión del archivo, al dar clic en Imprimir realiza la impresión correspondiente, si se da clic en Cancelar no realiza nada.

The 'Imprimir' dialog box is shown with the 'Preparar Página' tab selected. It contains the following settings:

- General:** Soporte (Support) section with 'Tamaño' (Size) set to 'Carta' and 'Origen' (Origin) set to 'Seleccionar Automáticamente'.
- Orientación:** Orientation section with four options: 'Vertical' (selected), 'Horizontal', 'Vertical Inverso', and 'Horizontal Inverso'.
- Márgenes:** Margins section with four input fields: 'izquierdo (mm)' (left) set to 25.4, 'derecho (mm)' (right) set to 25.4, 'superior (mm)' (top) set to 25.4, and 'inferior (mm)' (bottom) set to 25.4.
- Buttons:** 'Imprimir' (Print) and 'Cancelar' (Cancel) buttons at the bottom right.

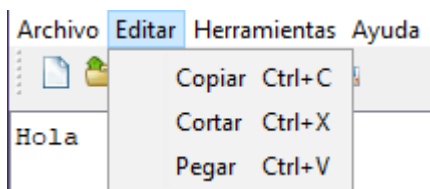
The 'Imprimir' dialog box is shown with the 'General' tab selected. It contains the following settings:

- Servicio de Impresión:** Print Service section with 'Nombre' (Name) set to 'HP Deskjet 2540 series (Red)', 'Estado' (Status) set to 'Aceptando trabajos', and 'Tipo' (Type) set to 'Información'. There is a 'Propiedades...' button and an 'Imprimir en Archivo' checkbox.
- Rango de Impresión:** Print Range section with 'Todo' (All) selected and 'Páginas' (Pages) set to '1' to '1'.
- Copias:** Copies section with 'Número de Copias' (Number of Copies) set to 1 and the 'Intercalar' (Interleave) checkbox checked.
- Buttons:** 'Imprimir' (Print) and 'Cancelar' (Cancel) buttons at the bottom right.

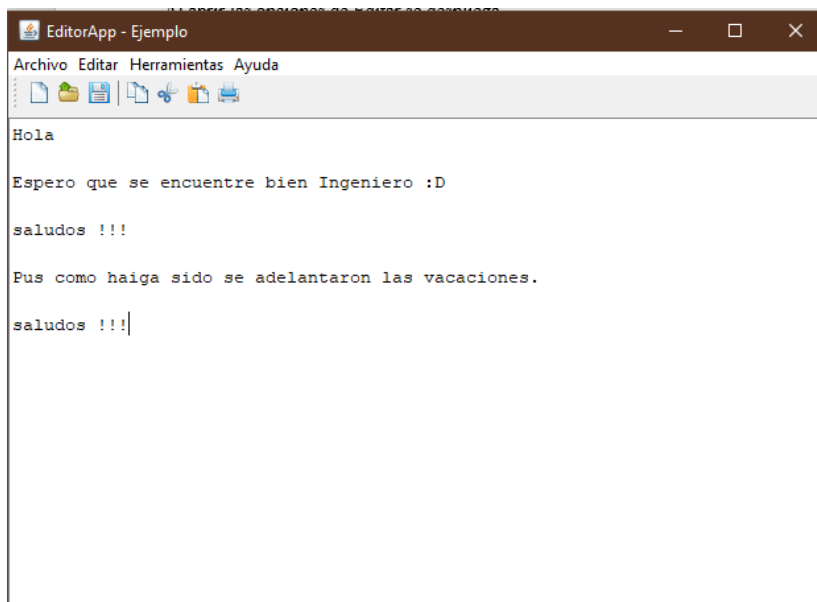


Al dar clic en Salir, se cierra la aplicación

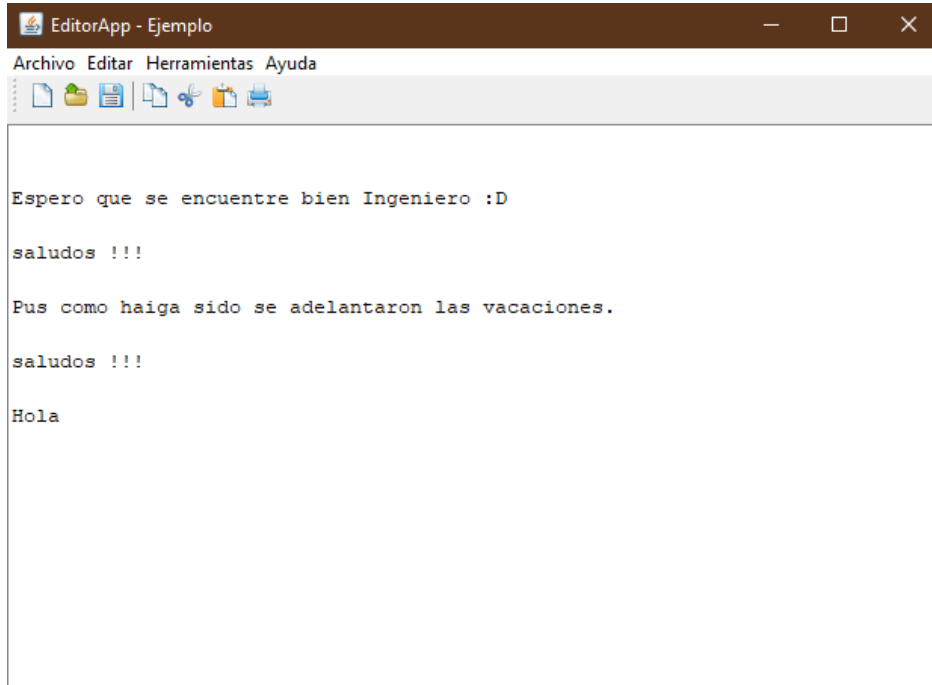
Al abrir las opciones de Editar se despliega



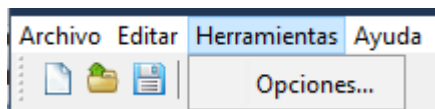
Al dar clic en Copiar, obtiene copiando el texto seleccionado y al dar clic en Pegar, plasma el texto copiado seleccionado previamente (en este caso fue seleccionado el texto de “saludos !!!”)



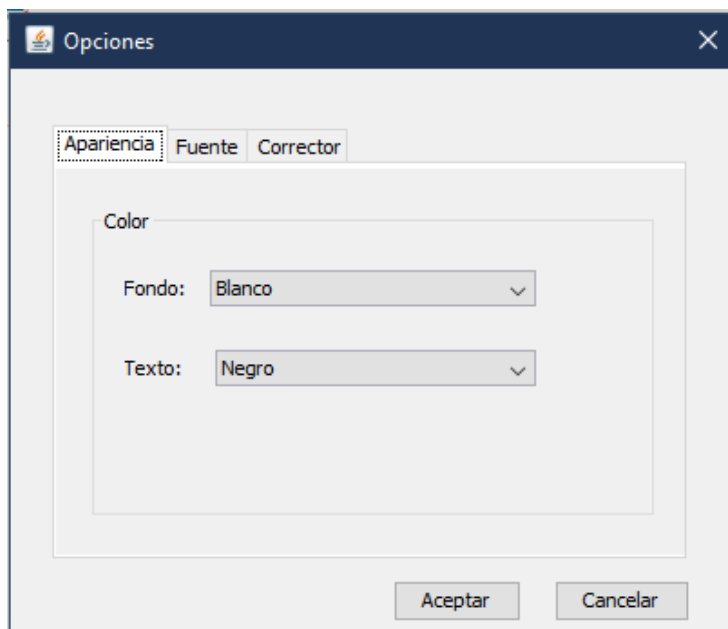
Al dar clic en Cortar, obtiene cortando el texto seleccionado y al dar clic en Pegar, plasma el texto cortado seleccionado previamente (en este caso fue el Hola)



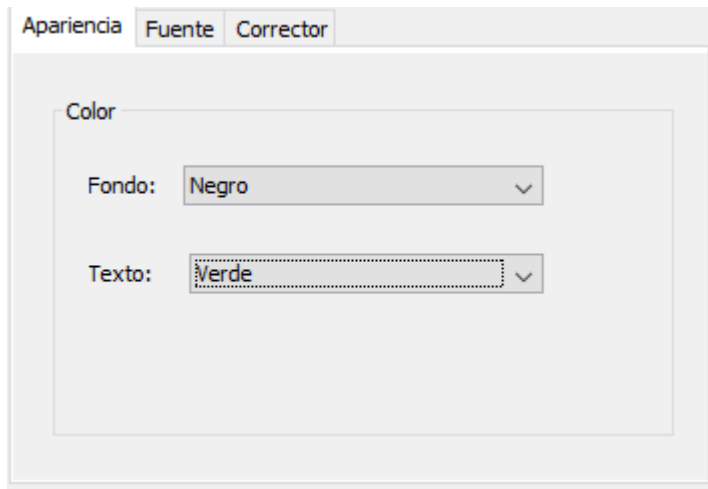
Al Abrir las opciones de Herramientas se despliega



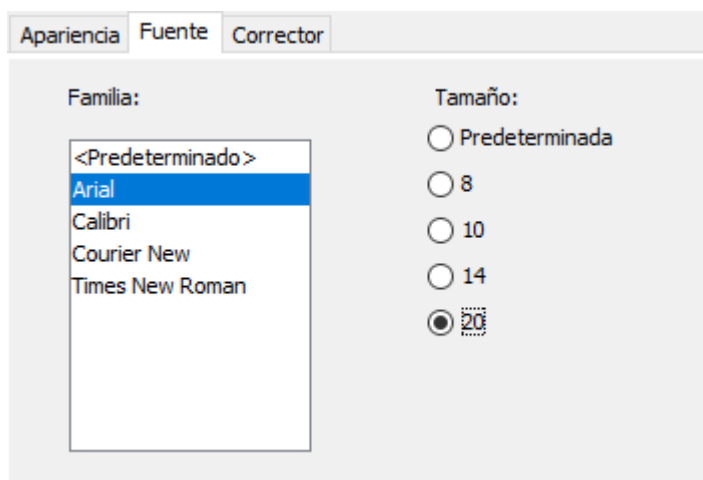
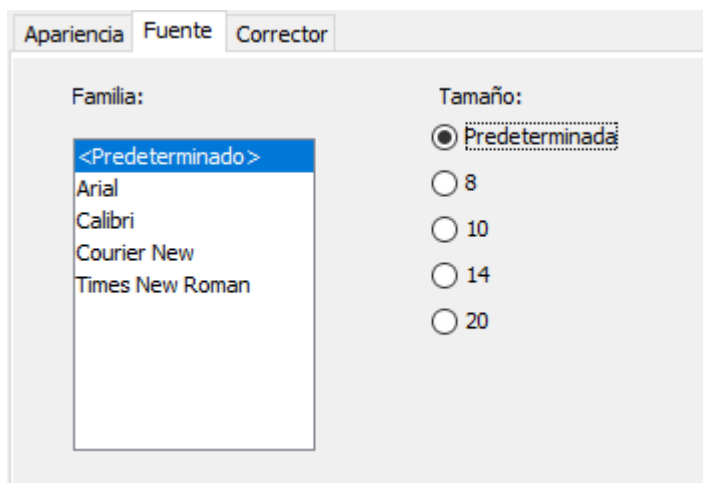
Al dar clic en Opciones se muestra un diálogo de Opciones en la que se compone de diferentes funciones para modificar la apariencia, fuente y un corrector de palabras, al dar clic en el botón de Aceptar se realizan los cambios y al dar clic en Cancelar no realiza nada, en la sección de Apariencia aparece así en default.



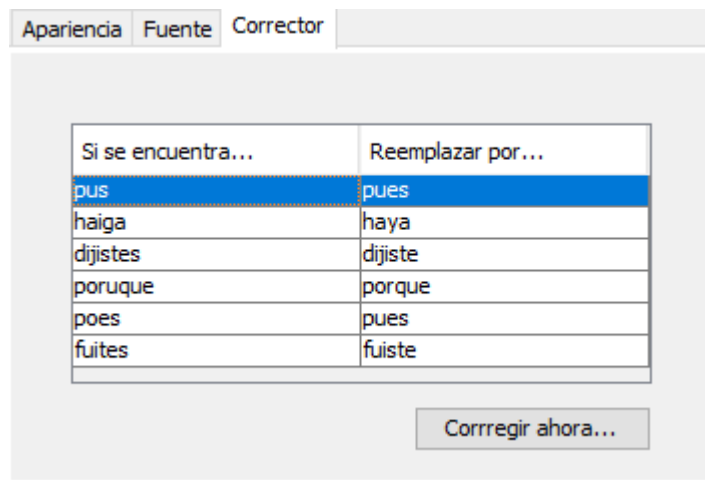
Aquí se cambia el color del fondo y del texto del editor, en este caso lo modificaremos para ver los cambios que realiza.



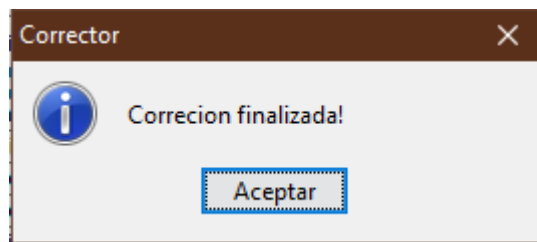
Aquí en la sección Fuente se cambia el tipo y tamaño de la fuente del editor, la primera imagen es en default la configuración y la segunda imagen es con una modificación



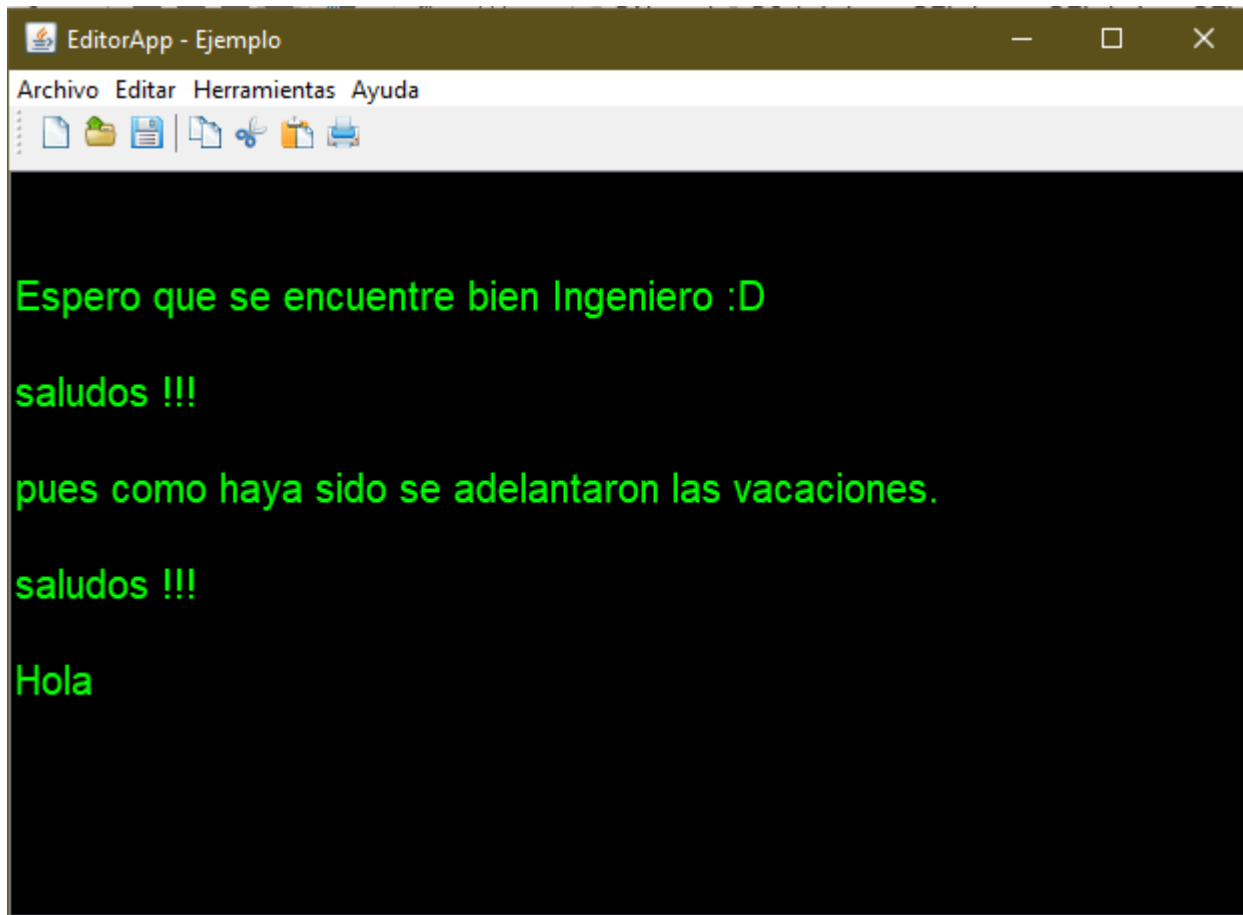
En la sección de Corrector, al dar clic en Corregir ahora se reemplazan las palabras que están mal ortográficamente (primera columna) con palabras bien escritas (segunda columna)



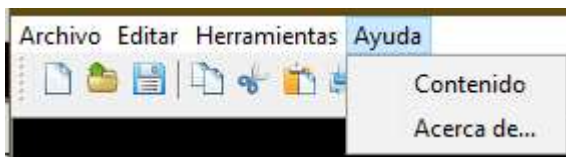
Al dar clic en Corregir ahora... aparece este dialogo con el siguiente mensaje:



Hacer clic en Aceptar en la ventana de Opciones para que se asuman las modificaciones realizadas anteriormente, y queda así el editor (se nota bastante diferencia).



Al Abrir las opciones de Ayuda se despliega



Al dar clic en Contenido se abre este mismo documento

Al dar clic en Acerca de... se abre una ventana de diálogo en la que se muestra información del autor de la aplicación.



La barra tiene las siguientes funciones ya explicadas anteriormente nombradas por orden:



Nuevo, Abrir, Guardar, Copiar, Cortar, Pegar, Imprimir.