

PRÁCTICA INTERFAZ GRÁFICA

Daniel Oliva González 44697438C



23 de marzo de 2019

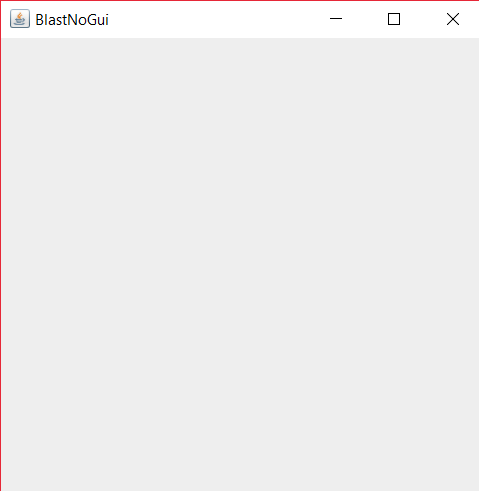
Universidad de málaga

# 1.Introducción y objetivo de la práctica

En esta práctica vamos a realizar una interfaz gráfica para el algoritmo BLAST, el cual nos permite realizar comparaciones entre secuencias que se encuentran almacenadas en una base de datos.

# 2. Creación de un JFrame

Para poder implementar la interfaz del algoritmo BLAST, vamos a construir un **JFrame** al que se le añadirán posteriormente diferentes paneles, botones y otros instrumentos que nos permitirán realizar lo requerido.

Así, el código para implementar nuestro JFrame sobre el que se construirá la interfaz gráfica será:

JFrame jfr=**new** JFrame("BlastNoGui");

jfr.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

Este código nos permitirá obtener este panel:

Como se puede ver el panel se encuentra vacío.

A continuación, vamos a ir añadiendo los elementos

necesarios.

Una vez añadamos el código de los paneles y sus respectivos

elementos, incluiremos los siguientes comandos que permitirán

la visualización del panel y tener un tamaño determinado

jfr.setVisible(**true**);

jfr.setSize(500, 500);

# 3. Creación de los JPanel

A continuación, crearemos dos JPanels para dividir nuestra interfaz.

## 3.1 Creación JPanel1

De este modo, crearemos un primer panel en el que se nos permitirá elegir el tipo de secuencia que se quiere buscar.

Panel p1=**new** JPanel();

p1.setLayout(**new** FlowLayout());

p1.setBackground(Color.***YELLOW***);

## 3.2 Creación de los JRadioButton

Para proseguir vamos a crear una etiqueta indicándole al usuario que escoja entre **PROTEÍNAS** y **NUCLEÓTIDOS**. Crearemos un tipo de fuente (que también nos servirá para más adelante) para la letra y así modificar el tipo de letra con el que se escribe en la etiqueta.

JLabel label=**new** JLabel("Escoja el tipo de secuencia: ");

label.setSize(20, 20);

Font fuente=**new** Font("TimesRoman", Font.***BOLD***, 20);

label.setFont(fuente);

Por último, crearemos los dos botones con sus respectivos nombres y los distribuiremos por el panel.

JRadioButton jBPro=**new** JRadioButton("PROTEÍNAS");

jBPro.setBackground(Color.***CYAN***);

jBPro.setBorderPainted(**true**);

JRadioButton jBNuc=**new** JRadioButton("NUCLEÓTIDOS");

jBNuc.setBackground(Color.***CYAN***);

jBNuc.setBorderPainted(**true**);

p1.add(label,BorderLayout.***WEST***);

p1.add(jBPro,BorderLayout.***EAST***);

p1.add(jBNuc,BorderLayout.***EAST***);

### 3.2.1 ActionListener de los botones

Además, tenemos que añadir un ActionListener, de tal forma que, si se presiona un botón, el otro no permanecerá marcado:

jBPro.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**if**(jBPro.isSelected()) {

jBNuc.setSelected(**false**);

}

}

});

jBNuc.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**if**(jBNuc.isSelected()) {

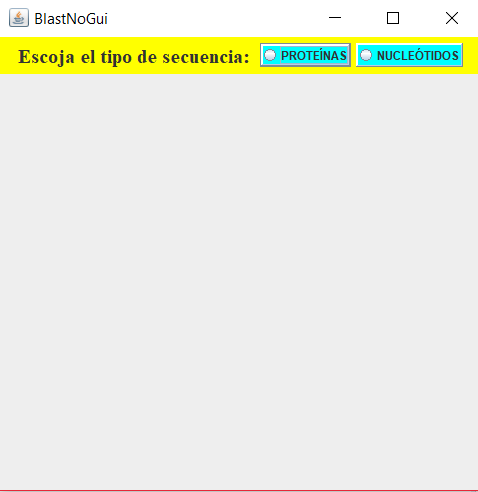
jBPro.setSelected(**false**);

}

}

});

Con esto, daríamos por concluido el primer panel con este resultado:



# 4. Creación del JPanel2

En este JPanel vamos a introducir diferentes elementos. Para ello utilizaremos los mismos comandos que en el panel anterior.

En este panel, nos encargaremos de introducir la secuencia que queremos buscar, el porcentaje de acierto y mostrar el resultado de la búsqueda.

## 4.1 Creación de JComboBox y JLabel de la secuencia

El primer elemento que vamos a introducir va a ser un JLabel y un ComboBox, donde escribiremos la secuencia que deseamos buscar.

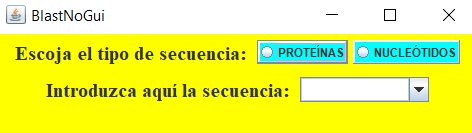
Primero crearemos la etiqueta:

JLabel j2=**new** JLabel("Introduzca aquí la secuencia: ");

j2.setFont(fuente);

Segundo, vamos a crear el combobox donde se introducirá la secuencia que se quiere buscar.

Finalmente, el resultado una vez se añada será el siguiente:



### 4.1.1 ActionListener del JComboBox

Añadiremos un ActionListener cuya función será que el usuario pueda recuperar la cadena ya introducida con anterioridad. El código será:

cSec.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

String seqq= cSec.getEditor().getItem().toString();

cSec.addItem(seqq);

}

});

## 4.2 Creación de JTextArea y JLabel del porcentaje de acierto

El segundo elemento que vamos a introducir va a ser un JTextArea junto a un JLabel. En este JLabel indicaremos que va el JTextArea vamos a escribir el porcentaje de acierto.

Para crear la etiqueta usaremos:

JLabel j3=**new** JLabel("Introduzca el porcentaje: ");

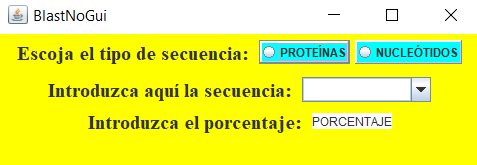
j3.setFont(fuente);

Y para crear el JTextArea donde introduciremos el porcentaje usaremos:

JTextArea porc=**new** JTextArea("PORCENTAJE",1,2);

porc.setEditable(**true**);

Así obtendremos como resultado:



## 4.3 Creación JTextArea, JButton y JScrollPane del resultado

En esta ocasión, crearemos un cuadro de texto que exponga el resultado una vez se haya escrito la secuencia y pulsado el botón que nos permite realizar la búsqueda.

Para facilitar la lectura del resultado, lo incluiremos en un JScrollPane.

Para crear el botón usaremos:

JButton bResul=**new** JButton("Pulse para resultado");

bResul.setBackground(Color.***CYAN***);

bResul.setBorderPainted(**true**);

Para la creación del cuadro de texto usaremos:

JTextArea jtResul=**new** JTextArea("Aquí aparecerá su resultado",20,28);

jtResul.setEditable(**false**);

En este caso, al cuadro de texto le establecemos un tamaño inicial grande ya que queremos que se nos muestren todos los resultados obtenidos. Además, seleccionamos que no se pueda editar.

Y para la creación de la barra:

JScrollPane jSec=**new** JScrollPane(jtResul);

El fin de la barra es poder ver todo el resultado además de facilitar la lectura. Para crearla tenemos que pasarle al constructor el objeto al que queremos que se le aplique el JScrollPane.

### 4.3.1 ActionListener del JButton

Este ActionListener se va a encargar de lanzarnos al resultado al cuadro de texto dependiendo de si buscamos una proteína o una secuencia de nucleótidos como se observa en el código. Además, mostrará un error por pantalla en el caso de no encontrar dicha secuencia.

bResul.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

String t="Proteins";

**if**(jBNuc.isSelected()) {

t="Nucleotids";

}

f**loat** porcentaje=Float.*parseFloat*(porc.getText());

String secuencia=cSec.getSelectedItem().toString();

**try** {

**if**(t=="Proteins") {

String r = bCnt.blastQuery('p', ***dataBaseFile***,***dataBaseIndexes***, porcentaje, secuencia);

jtResul.setText(r);

}**else** {

String r = bCnt.blastQuery('n', ***dataBaseFile***,***dataBaseIndexes***, porcentaje, secuencia);

jtResul.setText(r);

}

}**catch**(Exception a) {

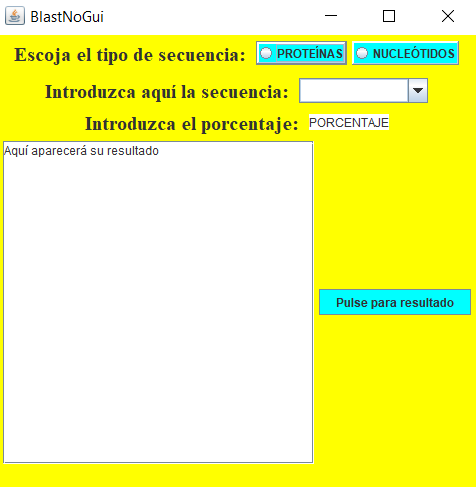
jtResul.setText(“No existen coincidencias”)

}

}

});

De esta manera el resultado obtenido será:



El enlace al repositorio de GitHub es: <https://github.com/danioliva/ISA1819BLAST-OLIVA>