



MODELO PARA DETERMINACIÓN DE LAS CONSTANTES CINÉTICAS DE LAS REACCIONES BIOLÓGICAS EN LAS AGUAS RESIDUALES DOMESTICAS

Gonzalo Villa Manosalvas¹

Docente Investigador de la Carrera Ingeniería Química de la Universidad de Guayaquil
gonzalo.villam@ug.edu.ec

José Cárdenas Murrillo²

Decano de la Facultad de Ingeniería Química de la Universidad de Guayaquil
Jose.cardenasm@ug.edu.ec

Teresa Villa Cox³

Docente de la Facultad de Ingeniería Química de la Universidad de Guayaquil
viviana.villa.c@ug.edu.ec

Para citar este artículo puede utilizar el siguiente formato:

Gonzalo Villa Manosalvas, José Cárdenas Murrillo y Teresa Villa Cox (2017): "Modelo para determinación de las constantes cinéticas de las reacciones biológicas en las aguas residuales domesticas", Revista Caribeña de Ciencias Sociales (agosto 2017). En línea: <http://www.eumed.net/rev/caribe/2017/08/aguas-residuales-domesticas.html>

Resumen

Los coeficientes biocinéticos necesarios para el diseño de reactores biológicos, se determinan a partir de resultados obtenidos en equipos pilotos de flujo continuo o discontinuo, aplicando el método diferencial o de integración. Utilizando las concentraciones remanentes, determinamos la variación del sustrato en función del tiempo, aplicando un método de tanteo se resolvió las expresiones de la velocidad y el orden de la reacción. Conocido este parámetro creamos una interface que resolvió los modelos matemáticos propuestos, permitiendo una comunicación con el usuario, la plataforma que se escogió para el desarrollo de la investigación es la GUI de Matlab, versión R-2015^a

Palabras claves:

Coefficiente cinético – ecuaciones – modelado - Método diferencial - GUI de Matlab

Abstract

The biokinetic coefficients necessary for the design of biological reactors are determined from results obtained in continuous or discontinuous flow pilots

using the differential or integration method. Using the remaining concentrations, we determined the variation of the substrate as a function of time, using a method of scoring, the expressions of the speed and order of the reaction were solved. Known this parameter we created an interface that solved the proposed mathematical models, allowing a communication with the user, the platform that was chosen for the research development is the Matlab GUI, version R-2015a

Keyword:

Kinetic coefficient - equations - modeling - Differential method - Matlab GUI

1. INTRODUCCIÓN

1.1. Desarrollo del método para ecuaciones de grado cero

La ecuación definida para velocidades de orden de reacción de grado n, se da por la siguiente expresión. (Martinez & Rodriguez, 2005)

$$-r_A = \frac{dC_A}{dt} = kC_A^n \quad (1)$$

Dónde:

CA = Concentración del reactante A.

k = Constante de la velocidad de reacción.

t = Tiempo.

n = Orden de reacción.

Con base en la ecuación de velocidad (1), podemos escribir para reacciones de orden cero (n = 0):

$$-r_A = \frac{dC_A}{dt} = kC_A^0 \quad (2)$$

Revisando la ecuación 2, vemos que las reacciones de orden 0 son una función lineal de la concentración, ya que al integrar la misma entre una concentración inicial C_{A0} , hasta una concentración final C_A , y luego arreglando los términos, obtenemos la ecuación de la recta ($y = mx+b$) de la forma siguiente:

$$C_A = C_{A0} - kt \quad (3)$$

Al graficar CA vs t obtenemos una recta con pendiente negativa igual a k.

1.2. Desarrollo del método para ecuaciones de primer grado

Para reacciones de primer orden (n = 1), del $A \rightarrow B$ tipo (Martinez & Rodriguez, 2005):

Seguimos el mismo procedimiento que las reacciones de orden cero y se obtiene la siguiente ecuación:

$$-r_A = \frac{dC_A}{dt} = kC_A^1 \quad (4)$$

Si arreglamos la ecuación 4 e integramos entre una concentración inicial C_{A0} , hasta una concentración final C_A , obtenemos:

$$-\ln \frac{C_A}{C_{A0}} = k_1 t \quad (5)$$

$$-\ln \frac{C_A}{C_{A0}} \text{ vs } t \quad (6)$$

Podemos graficar:

Para obtener una recta que tiene como pendiente la constante de velocidad k_1 .

Desarrollo del método para ecuaciones de segundo grado

Para reacciones de segundo orden ($n = 2$), del $A + A \rightarrow B$ tipo:

Seguimos el mismo procedimiento que las reacciones de orden cero y se obtiene la siguiente ecuación:

$$-r_A = \frac{dC_A}{dt} = kC_A^2 \quad (7)$$

Si arreglamos la ecuación 7 e integramos entre una concentración inicial C_{A0} , hasta una concentración final C_A , obtenemos la ecuación de una recta.

$$\frac{1}{C_A} = kt + \frac{1}{C_{A0}} \quad (8)$$

Si graficamos $1/C_A$ vs t , podemos obtener la pendiente de la recta (k) y la ordenada al origen ($1/C_{A0}$). Es importante anotar que en las reacciones Biológicas no se suelen dar reacciones de segundo grado.

1.3. Desarrollo del método para ecuaciones de tipo de saturación (cinética de orden variable)

Dentro de las reacciones biológicas algunas se comportan de forma que su ecuación de velocidad toma la forma de:

$$-r_A = \frac{dC_A}{dt} = \frac{k_1 C_A}{1 + k_2 C_A} \quad (9)$$

En este tipo de ecuaciones se suelen dar casos particulares como por ejemplo a concentraciones altas $K_2 * C_A \gg 1$; por tanto, en el denominador el término $1 + K_2 * C_A$ es aproximadamente igual a $K_2 * C_A$ por tanto la ecuación 9 se la puede simplificar de la siguiente manera.

$$-r_A = \frac{dC_A}{dt} = \frac{k_1 \cancel{C_A}}{k_2 \cancel{C_A}} \approx \frac{k_1}{k_2} = k \quad (10)$$

Este comportamiento es parecido a la cinética de orden cero. En cambio cuando se alcanzan bajas concentraciones, es decir cuando $1 \gg K_2 * C_A$ entonces.

$$1 + k_2 C_A \approx 1 \quad (11)$$

Con lo cual la ecuación de la velocidad 7, se transforma:

$$-r_A = \frac{dC_A}{dt} = \frac{k_1 C_A}{1 + k_2 C_A} \approx \frac{k_1 C_A}{1} = k_1 C_A \quad (12)$$

Notamos que para estas condiciones su comportamiento es similar a una ecuación de primer orden, de ahí el término cinéticas de orden variable. Para encontrar las constantes de este tipo de ecuaciones, se suele obtener el inverso de la ecuación 9

$$\frac{1}{-r_A} = \frac{1 + k_2 C_A}{k_1 C_A} = \frac{1}{k_1 C_A} + \frac{k_2 C_A}{k_1 C_A} \quad (13)$$

Que es la ecuación de una recta como podemos notar: $\frac{1}{-r_A} = \frac{1}{k_1 C_A} + \frac{k_2}{k_1}$ (14)

Se grafica $1/-r_A$ vs $1/C_A$, para obtener una recta con pendiente $1/k_1$, y una ordenada al origen k_2/k_1 .

2. MATERIALES Y MÉTODOS

2.1. Plataforma de desarrollo

Para el desarrollo de este modelo utilizaremos Matlab (abreviatura de MATrix LABoratory, o "Laboratorio de Matrices"). Entre las muy extensas prestaciones del programa encontramos solo para nombrar unas cuantas la manipulación de matrices, el cálculo numérico y el cálculo simbólico, su gran capacidad gráfica, la implementación de algoritmos, la implementación de interfaces de usuario (GUI), su capacidad e comunicarse con otros programas (Excel, por ejemplo), su capacidad de comunicarse con otros lenguajes de programación (C, C++, VB, Java); su capacidad de comunicarse con hardware y a través de ellos llevara control de equipos. Como herramientas de apoyo a Matlab encontramos Simulink y la GUIDE de Matlab (editor de interfaces de usuario - GUI).

2.2. Guide interfaz gráfica de usuario en Matlab

El uso de las herramientas que proporciona la GUIDE de Matlab permite desarrollar interfaces de usuarios para creación de Apps personalizadas que permiten el manejo solo con el mouse de modelos matemáticos de alto perfil, sin tener que acudir a lenguajes de programación tradicionales como visual basic, C++ o el mismo lenguaje de programación de Matlab. (MathWorks, 2017).

Utilizando el editor de diseño de GUIDE, se crea la aplicación, con solo arrastrar y soltar los objetos de la Paleta de Componentes al Área de Diseño, cada uno de los objetos tiene una función asociada donde se escribe el código que va a ser ejecutado por el objeto.

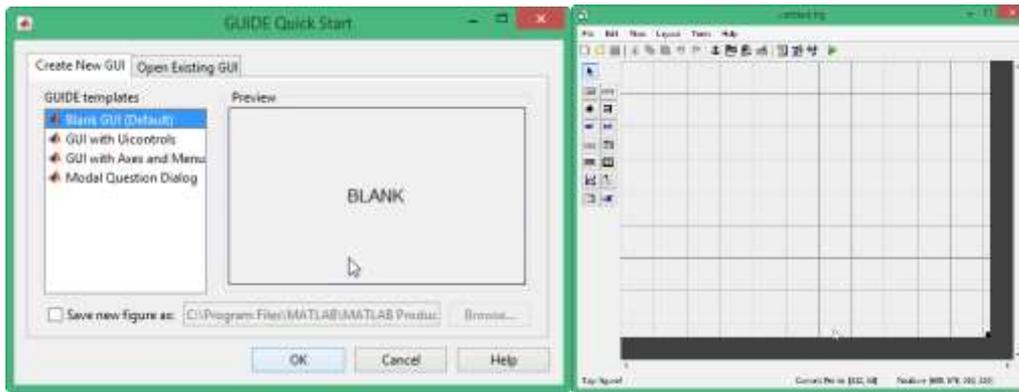


Figura 1.- Guide Quick Start y la Ventana de Diseño

3. METODOLOGIA USADA PARA EL DESARROLLO DEL MODELO PARA DETERMINACIÓN DE LAS CONSTANTES CINÉTICAS DE LAS REACCIONES BIOLÓGICAS EN LAS AGUAS RESIDUALES DOMESTICAS

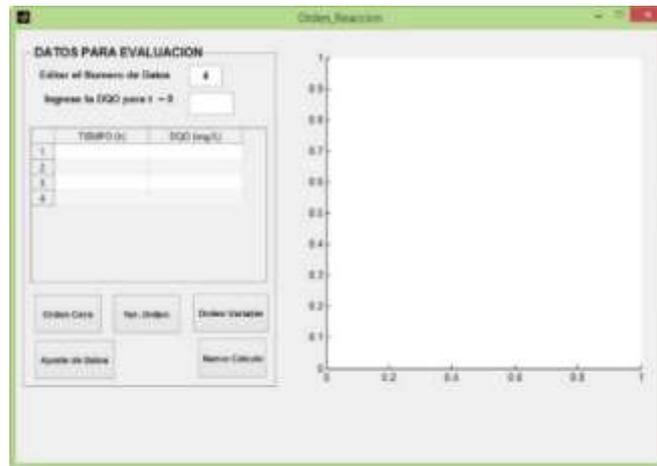


Figura 2. Formulario para la Determinación de las Constantes Cinéticas.

Como ejemplo de uso del modelo vamos utilizar los datos presentados en la tabla 1 que fueron determinados en un reactor por lotes, para determinar el orden de reacción del proceso biológico y los coeficientes cinéticos.

Tabla 1. Datos para el modelado cinético

Tiempo (d)	Concentración [C] (mol/L)
0	100.00
1	71.2
2	51.6
3	37
4	25.6

5	19.9
6	13.1
7	9.5
8	6.9

Fuente: (Tchobanoglous, 2000)

Para resolver este problema utilizamos el modelo propuesto y al momento de pulsar el icono de arranque del modelo la Ayuda del mismo presenta un mensaje de sistema para empezar a usar de manera correcta el mismo.

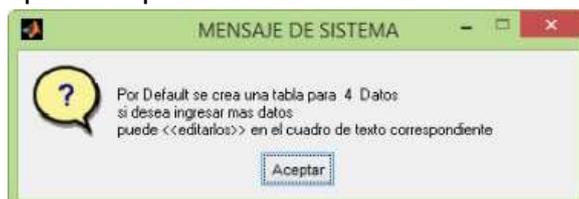


Figura 3. Mensaje de Sistema

Vemos que el mensaje nos informa que el modelo crea por defecto una tabla para 4 datos, para presentar las gráficas X-Y, si usted necesita ingresar más datos debe editarlos en el cuadro de texto llamado <<Editar el Numero de datos>>.



Figura 4. Ingreso del número de datos

En este caso tenemos un dato para tiempo = 0, y 8 datos para diversos valores de tiempo, por tanto, ingresamos los datos en sus lugares correspondientes. Primero creamos la tabla de datos necesaria ingresando el número 8 y pulsando tecla <<ENTER>>.

Se forma la tabla Tiempo vs DQO con 8 entradas listas, donde procedemos a ingresar los datos de la Tabla 1. Es importante saber que si pulsamos cualquiera de los <<Botones de Acción>> del modelo antes de ingresar datos, este nos responderá con un mensaje de error. Por ejemplo, si no hemos ingresado los datos de la Tabla 1 veremos el siguiente mensaje.

Usted deberá pulsar el botón <<OK>>, para regresar al formulario e ingresar los datos correspondientes. No olvide que hay un cuadro de texto especial para ingresar los datos del DQO (Concentración) para tiempo 0. Si usted no ingresa ese dato en el cuadro apropiado (lo ingresa en la tabla o simplemente olvida ingresarlo el sistema le responderá con el siguiente mensaje).



Figura 5. Mensaje de Sistema, que controla el ingreso de C_0

Una vez que usted ha ingresado los datos de manera correcta está listo a presentar estos datos de forma gráfica, el modelo le ofrece varios botones para graficar los datos de acuerdo con los siguientes órdenes de reacción sugeridos.

Tiene tres opciones a saber:

- Orden Cero.
- Primer Orden
- Orden Variable

Usted podría seleccionar estos botones de manera secuencial y visualizar con cual obtiene aproximadamente una línea recta. Presentaremos los resultados obtenidos con cada uno de los botones anotados. Los pares ordenados de la tabla 1 para el caso de una gráfica de Orden Cero presentan una curva, por tanto, los datos no reflejan una reacción de este orden. Seguidamente, pulsamos el siguiente botón <<Orden 1>> que nos entrega la siguiente gráfica.

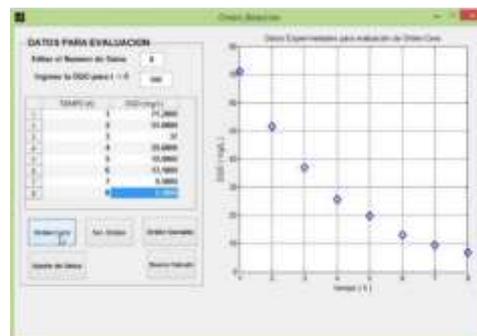
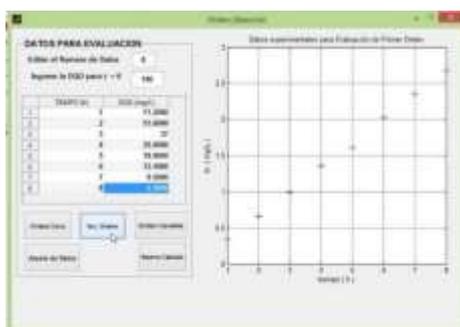


Figura 6. Análisis para reacciones de 1er Orden Figura 7. Análisis para reacciones de Orden 0

Vemos que obtenemos una recta con los datos de la Tabla 1 por tanto podemos plantear que estos datos tienen el comportamiento cinético de una reacción de primer orden. Finalmente, solo a manera de comprobación pulsamos el tercer botón, para obtener la siguiente gráfica.

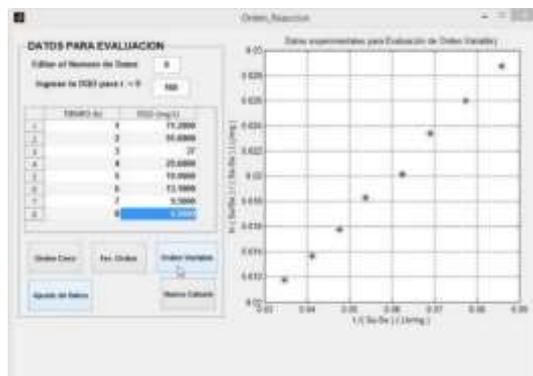


Figura 8. Análisis para reacciones de Orden Variable

Podemos anotar que los puntos tienen una ligera desviación de una recta en el sexto punto, ahora para comprobar nuestra apreciación pulsamos el botón <<AJUSTE DE DATOS>>, que nos envía un mensaje de sistema de la siguiente manera.



Figura 9. Mensaje de sistema para Linealizar los puntos

Si pulsa el botón <<NO>>; regresará al formulario del modelo y podrá verificar si desea editar cualquier dato. En caso de que desee linealizar los puntos del grafico entonces deberá pulsar <<SI>> en el mensaje de sistema, para obtener los <<Parámetros de Ajuste>>.



Figura 10. Selección de Parámetros de ajuste

Note que dentro de la sección <<Parámetros de Ajuste>>; encontramos un cuadro combinado llamado <<Linealizar los Datos>>; que nos ofrece las posibilidades de obtener una recta que pase por el mayor número de puntos posible, con el menor error de desviación posible, para cada uno de los órdenes de reacción que nos ofrece el sistema y además las constantes de velocidad de reacción se pueden obtener de estas rectas las mismas que se

presentan en los cuadros de texto llamados <<Constante K1>>; y/o según sea el caso <<Constante K2>>.

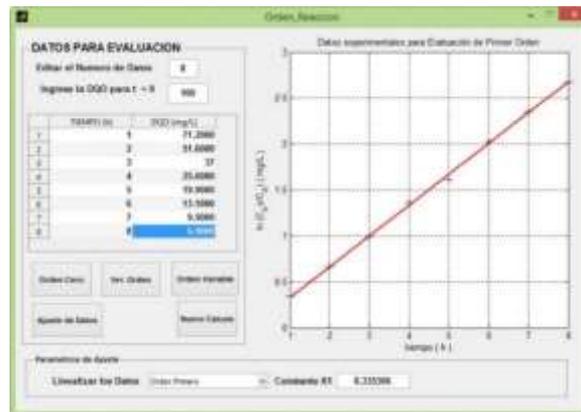


Figura 11. Línea ajustada y constante Cinética reportada

El modelo nos reporta la constante buscada y el grado de ajuste de la reacción cinética, en este caso particular tenemos que: $K_1 = 0.335306$ y el orden de la reacción es primero. Por tanto, la ecuación cinética según la ec. 4 quedaría

$$-r_A = \frac{dC_A}{dt} = 0.335306 C_A \quad (15)$$

Si arreglamos la ecuación 15 e integramos entre una concentración inicial C_{A0} , hasta una concentración final C_A , obtenemos:

$$-\ln \frac{C_A}{C_{A0}} = 0.335306 t \quad (16)$$

3.1. Código para el modelo orden de reacción

Las filas del código están numeradas de manera secuencial, varios números de línea del código han sido eliminados porque son líneas vacías que se dejan para comentarios y/o código complementario

```

1 function varargout = Orden_Reaccion(varargin)
2 % ORDEN_REACCION MATLAB code for Orden_Reaccion.fig
25 % Last Modified by GUIDE v2.5 29-Mar-2017 10:11:24
28 gui_Singleton = 1;
29 gui_State = struct('gui_Name',    mfilename, ...
30     'gui_Singleton',  gui_Singleton, ...
31     'gui_OpeningFcn', @Orden_Reaccion_OpeningFcn, ...
32     'gui_OutputFcn',  @Orden_Reaccion_OutputFcn, ...
33     'gui_LayoutFcn',  [], ...
34     'gui_Callback',   []);
35 if nargin && ischar(varargin{1})
36     gui_State.gui_Callback = str2func(varargin{1});

```

```

37 end
39 if nargin
40     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
41 else
42     gui_mainfcn(gui_State, varargin{:});
43 end
46 % --- Executes just before Orden_Reaccion is made visible.
47 function Orden_Reaccion_OpeningFcn(hObject, eventdata, handles, varargin)
48 % This function has no output args, see OutputFcn.
55 handles.output = hObject;
57 % Update handles structure
58 guidata(hObject, handles);
63 % --- Outputs from this function are returned to the command line.
64 function varargout = Orden_Reaccion_OutputFcn(hObject, eventdata, handles)
65 % varargout cell array for returning output args (see VARARGOUT);
71 varargout{1} = handles.output;
73 % --- Executes on button press in Orden_Uno.
74 function Orden_Uno_Callback(hObject, eventdata, handles)
75 % hObject handle to Orden_Uno (see GCBO)
79 set(handles.Ajuste_Recta,'Visible','off')
81 gonzalo = get(handles.uitable1,'Data');
82 a=cell2mat(gonzalo);
83 if numel(a)==0
84     errordlg('Debe usted ingresar datos en la tabla','MENSAJE DE SISTEMA')
85     return
86 end
87 aaa=str2double(get(handles.DQO_uni,'String'));
89 if isnan(aaa)
90     errordlg('Debe usted ingresar datos para el DQO a tiempo 0','MENSAJE DE SISTEMA')
91     return
92 end
94 X=a(:,1);
95 DQO=a(:,2);
96 So = str2num(get(handles.DQO_uni,'String'));
98 Y = log(So./DQO); % introducir ./ para realizar division con el vector OQO
99 plot(X, Y,'+ k','markersize',10)
100 title('Datos experimentales para Evaluación de Primer Orden')
101 xlabel('tiempo ( h )')
102 ylabel('ln ( mg/L )')
103 grid on
105 % --- Executes on button press in Orden_Variable.
106 function Orden_Variable_Callback(hObject, eventdata, handles)
107 % hObject handle to Orden_Variable (see GCBO)
111 set(handles.Ajuste_Recta,'Visible','off')
113 gonzalo = get(handles.uitable1,'Data');
114 a=cell2mat(gonzalo);
115 if numel(a)==0
116     errordlg('Debe usted ingresar datos en la tabla','MENSAJE DE SISTEMA')
117     return
118 end
119 aaa=str2double(get(handles.DQO_uni,'String'));
120 if isnan(aaa)

```

```

121 errordlg('Debe usted ingresar datos para el DQO a tiempo 0','MENSAJE DE SISTEMA')
122 return
123 end
124 t= a(:,1);
125 So = str2num(get(handles.DQO_uni,'String'));
126 DQO= a(:,2);
128 X1 = t./(So-DQO);
129 Y1 = log(So./DQO)./(So-DQO);
130 plot(X1, Y1,'*k','markersize',10)
131 grid on
132 title('Datos experimentales para Evaluación de Orden Variable')
133 xlabel('t / ( So-Se ) ( Lh/mg )')
134 ylabel('ln ( So/Se ) / ( So-Se ) ( L/mg )')
135 text (5,4.5, ' * Datos Experimentales')
137 % --- Executes on button press in Orden_Cero.
143 function DQO_uni_Callback(hObject, eventdata, handles)
151 % --- Executes during object creation, after setting all properties.
152 function DQO_uni_CreateFcn(hObject, eventdata, handles)
153 % hObject handle to DQO_uni (see GCBO)
158 if ispc && isequal(get(hObject,'BackgroundColor'), get
(0,'defaultUicontrolBackgroundColor'))
159 set(hObject,'BackgroundColor','white');
160 end
163 % --- Executes on button press in Nuevo_Calculo.
164 function Nuevo_Calculo_Callback(hObject, eventdata, handles)
165 % hObject handle to Nuevo_Calculo (see GCBO)
16/06/14 11:45 PM F:\TESIS MAESTR...\Orden_Reaccion.m 5 of 13
166 % eventdata reserved - to be defined in a future version of MATLAB
169 a=get(handles.uitable1,'Data');
170 [r,c]=size(a);
171 set(handles.uitable1,'Data',cell(r,c));
172 set(handles.DQO_uni,'String','')
173 axes(handles.axes1);
174 W=[1];
175 Z=[1];
176 plot(W,Z)
178 % --- Executes on button press in Ajuste.
179 function Ajuste_Callback(hObject, eventdata, handles)
180 % hObject handle to Ajuste (see GCBO)
184 gonzalo = get(handles.uitable1,'Data');
185 a=cell2mat(gonzalo);
186 if numel(a)==0
187 errordlg('Debe usted ingresar datos en la tabla','MENSAJE DE SISTEMA')
188 return
189 end
190 aaa=str2double(get(handles.DQO_uni,'String'));
191 if isnan(aaa)
192 errordlg('Debe usted ingresar datos para el DQO a tiempo 0','MENSAJE DE SISTEMA')
193 return
194 end
196 Respuesta=questdlg('Desea Linealizar los Puntos....','INFORMACION DE
SISTEMA','Si','No','Si');
197 if strcmp(Respuesta,'Si')

```

```

198     set(handles.Ajuste_Recta,'Visible','on');
199     elseif strcmp(Respuesta,'No')
200         set(handles.Ajuste_Recta,'Visible','of');
201     end
204 % --- Executes during object creation, after setting all properties.
205 function Ajuste_CreateFcn(hObject, eventdata, handles)
206 % hObject    handle to Ajuste (see GCBO)
211 function Orden_Reaccion_CreateFcn(hObject, eventdata, handles)
212 % hObject    handle to Orden_Reaccion (see GCBO)
216 Respuesta = questdlg('Por Default se crea una tabla para 4 Datos', 'si desea ingresar
mas datos', ...
217 {'puede <<editarlos>> en el cuadro de texto correspondiente'}, 'MENSAJE DE
SISTEMA', 'Aceptar', 'Aceptar');
220 % --- Executes during object creation, after setting all properties.
221 function Ajuste_Recta_CreateFcn(hObject, eventdata, handles)
222 % hObject    handle to Ajuste_Recta (see GCBO)
227 function nFilas_Callback(hObject, eventdata, handles)
228 % hObject    handle to nFilas (see GCBO)
235 r=str2double(get(hObject,'String'));
236 set(handles.uitable1,'Data',cell(r,2));
238 % --- Executes during object creation, after setting all properties.
239 function nFilas_CreateFcn(hObject, eventdata, handles)
240 % hObject    handle to nFilas (see GCBO)
246 if ispc && isequal(get(hObject,'BackgroundColor'), get
(0,'defaultUicontrolBackgroundColor'))
247     set(hObject,'BackgroundColor','white');
248 end
250 % --- Executes when entered data in editable cell(s) in uitable1.
251 function uitable1_CellEditCallback(hObject, eventdata, handles)
252 % hObject    handle to uitable1 (see GCBO)
263 function uitable1_DeleteFcn(hObject, eventdata, handles)
264 % hObject    handle to uitable1 (see GCBO)
268 function Cte_K1_Callback(hObject, eventdata, handles)
269 % hObject    handle to Cte_K1 (see GCBO)
278 function Cte_K1_CreateFcn(hObject, eventdata, handles)
279 % hObject    handle to Cte_K1 (see GCBO)
286 if ispc && isequal(get(hObject,'BackgroundColor'), get
(0,'defaultUicontrolBackgroundColor'))
287     set(hObject,'BackgroundColor','white');
288 end
291 function Cte_K2_Callback(hObject, eventdata, handles)
292 % hObject    handle to Cte_K2 (see GCBO)
302 function Cte_K2_CreateFcn(hObject, eventdata, handles)
303 % hObject    handle to Cte_K2 (see GCBO)
311 if ispc && isequal(get(hObject,'BackgroundColor'), get
(0,'defaultUicontrolBackgroundColor'))
312     set(hObject,'BackgroundColor','white');
313 end
317 % --- inactive control, or over an axes background.
318 function Orden_Reaccion_WindowButtonDownFcn(hObject, eventdata, handles)
319 % hObject    handle to Orden_Reaccion (see GCBO)
325 function Menu_Orden_Callback(hObject, eventdata, handles)
326 % hObject    handle to Menu_Orden (see GCBO)

```

```

330 % Hints: contents = cellstr(get(hObject,'String')) returns Menu_Orden contents as cell array
331 %     contents{get(hObject,'Value')} returns selected item from Menu_Orden
333 Orden = get(handles.Menu_Orden,'Value');
335 switch Orden
337     case 2 %Orden Cero
339         set(handles.Cte_K2,'Visible','off')
340         set(handles.K2,'Visible','off')
341         gonzalo = get(handles.uitable1,'Data');
342         a=cell2mat(gonzalo);
343         if numel(a)==0
344             errordlg('Debe usted ingresar datos en la tabla','MENSAJE DE SISTEMA')
345             return
346         end
347         aaa=str2double(get(handles.DQO_uni,'String'));
348         if isnan(aaa)
349             errordlg('Debe usted ingresar datos para el DQO a tiempo 0','MENSAJE DE
SISTEMA')
350             return
351         end
353         axes(handles.axes1);
354         plot(a(:,1),a(:,2),'*', 'markersize',10)
355         grid on
356         hold on
358         % Obtencion de la recta que se ajusta mejor a los datos experimentales
360         X1=a(:,1);
361         Y1=a(:,2);
362         A = polyfit (X1,Y1, 1);
363         k1 = A(1); % pendiente k1
364         k2e = A(2);
365         k2 = -A(2); % ordenada al origen k2;
366         % Grafica de la recta que se ajusta a los datos experimentales
367         Ajuste = X1.*k1+k2e;
368         axes(handles.axes1);
369         plot(X1 ,Ajuste,'-b','LineWidth',2)
370         title ('Datos Experimentales para evaluación de Orden Cero')
371         xlabel('tiempo ( h )')
372         ylabel('DQO ( mg/L )')
373         text(2,75, '* * * *  Datos experimentales')
374         text(2, 50, '-----  Ajuste de Datos')
375         grid on
376         set(handles.Cte_K1,'String',k1)
377         hold off
379     case 3 %Orden Primero
381         set(handles.Cte_K2,'Visible','off')
382         set(handles.K2,'Visible','off')
383         gonzalo = get(handles.uitable1,'Data');
384         a=cell2mat(gonzalo);
385         if numel(a)==0
386             errordlg('Debe usted ingresar datos en la tabla','MENSAJE DE SISTEMA')
387             return
388         end
389         aaa=str2double(get(handles.DQO_uni,'String'));
390         if isnan(aaa)

```

```

391     errordlg('Debe usted ingresar datos para el DQO a tiempo 0','MENSAJE DE
SISTEMA') 392     return
393 end
395 X=a(:,1);
396 DQO=a(:,2);
397 So = str2num(get(handles.DQO_uni,'String'));
399 Y = log(So./DQO); % introducir ./ para realizar division con el vector OQO
400 plot(X, Y,'+k','markersize',10)
401 grid on
402 hold on
404 %Linealizar la curva
405 X1=a(:,1);
406 DQO=a(:,2);
407 So = str2num(get(handles.DQO_uni,'String'));
408 Y1 = log(So./DQO); % introducir ./ para realizar division con el vector OQO
410 % Obtencion de la recta que se ajusta mejor a los datos experimentales
411 A = polyfit (X1,Y1, 1);
412 k1 = A(1); % pendiente k1
413 k2 = A(2); % ordenada al origen k2
415 % Grafica de la recta que se ajusta a los datos experimentales
416 Ajuste = X1.*k1+k2;
417 plot(X1,Ajuste,'-r','LineWidth',2)
418 text(1, 5, '++++ Datos experimentales')
419 text(1, 4.5, '----- Ajuste de Datos')
420 title('Datos experimentales para Evaluación de Primer Orden')
421 xlabel('tiempo ( h )')
422 ylabel('ln (C_Ao/C_A) ( mg/L )')
423 grid on
424 set(handles.Cte_K1,'String',k1)
425 hold off
427 case 4 %Orden Variable
429 set(handles.Cte_K2,'Visible','on')
430 set(handles.K2,'Visible','on')
431 gonzalo = get(handles.uitable1,'Data');
432 a=cell2mat(gonzalo);
433 if numel(a)==0
434     errordlg('Debe usted ingresar datos en la tabla','MENSAJE DE SISTEMA')
435     return
436 end
437 aaa=str2double(get(handles.DQO_uni,'String'));
438 if isnan(aaa)
439     errordlg('Debe usted ingresar datos para el DQO a tiempo 0','MENSAJE DE
SISTEMA')
440     return
441 end
443 t= a(:,1);
444 So = str2num(get(handles.DQO_uni,'String'));
445 DQO= a(:,2);
446 X1 = t./(So-DQO);
447 Y1 = log(So./DQO)./(So-DQO);
448 plot(X1, Y1,'*k','markersize',10)
449 grid on
450 hold on

```

```

452 % Obtencion de la recta que se ajusta mejor a los datos experimentales
454 A = polyfit (X1,Y1, 1);
455 k1 = A(1); % pendiente k1
456 k2e = A(2);
457 k2 = -A(2); % ordenada al origen k2;
458 % Grafica de la recta que se ajusta a los datos experimentales
459 Ajuste = X1.*k1+k2e;
460 plot(X1 ,Ajuste,'r','LineWidth',2)
461 grid on
462 title('Datos experimentales para Evaluación de Orden Variable')
463 xlabel('t / ( So-Se ) ( L h / mg )')
464 ylabel('ln ( So/Se ) / ( So-Se ) ( L / mg )')
465 text(0.053,0.0095,'* * * * * Datos Experimentales')
466 text(0.053,0.0085,'----- Ajuste de Datos')
468 set(handles.Cte_K1,'String',k1)
469 set(handles.Cte_K2,'String',k2)
470 hold off
472 end
475 % --- Executes during object creation, after setting all properties.
476 function Menu_Orden_CreateFcn(hObject, eventdata, handles)
477 % hObject handle to Menu_Orden (see GCBO)
483 if ispc && isequal(get(hObject,'BackgroundColor'), get
(0,'defaultUicontrolBackgroundColor'))
484 set(hObject,'BackgroundColor','white');
485 end
487 % --- Executes on button press in Orden_Cero.
488 function Orden_Cero_Callback(hObject, eventdata, handles)
489 % hObject handle to Orden_Cero (see GCBO)
493 gonzalo = get(handles.uitable1,'Data');
494 a=cell2mat(gonzalo);
495 if numel(a)==0
496 errordlg('Debe usted ingresar datos en la tabla','MENSAJE DE SISTEMA')
497 return
498 end
499 aaa=str2double(get(handles.DQO_uni,'String'));
500 if isnan(aaa)
501 errordlg('Debe usted ingresar datos para el DQO a tiempo 0','MENSAJE DE SISTEMA')
502 return
503 end
505 axes(handles.axes1);
506 plot(a(:,1),a(:,2),'d','LineWidth',2,'markersize',10) 507 grid on
508 title ('Datos Experimentales para evaluación de Orden Cero')
509 xlabel('tiempo ( h )')
510 ylabel('DQO ( mg/L )')

```

4. CONCLUSIONES

La incorporación de lenguajes de programación matemática que favorecen el desarrollo de interfaces para comunicación con los usuarios. Nos permiten desarrollar aplicaciones muy útiles en el campo del tratamiento biológicos de aguas residuales, como las propuestas en este trabajo para el cálculo de las

constantes biológicas. La facilidad que aporta el uso de un formulario para realizar cálculos o presentar graficas hace que los usuarios aprovechen los resultados obtenidos sin tener que preocuparse por programar para obtener resultados.

El código abierto también es otro plus que se entrega en este trabajo puesto que esto permite a los lectores revisar la construcción del modelo, y hacer aportes para mejorarlo o tomar aquella parte del código que le sea útil para aplicaciones personalizadas

5. BIBLIOGRAFÍA

Martinez, S., & Rodriguez, M. (2005). *Tratamiento de Aguas Residuales con Matlab*. Mexico DF: Reverte.

MathWorks, T. (25 de junio de 2017). *GUI de Matlab*. Obtenido de <https://es.mathworks.com/discovery/matlab-gui.html>

Tchobanoglous, C. &. (2000). Destino de los Constituyentes de las Aguas residuales en el ambiente. En C. &. Tchobanoglous, *Tratamiento de Aguas Residuales en Pequeñas Poblaciones* (pág. 115). Bogota: Mc Graw-Hill.