ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

Materia: MÉTODOS NUMÉRICOS

PROYECTO BIMESTRAL

TEMA: Población

Integrantes:

- Leandro Bravo
- Juan Flores
- Stalin Jitala
- Brayan Ortiz

Quito, 16 de agosto de 2024

Contenido

Tabla de Figuras

OBJETIVOS

- Investigue sobre puntos fijos e iteración funcional.
- Determine la expresión matemática del valor cual converge la población cuando $1 \leq f \leq 2$.
- Determine el número y bifurcaciones para todos los valores de $f \geq 0$.
- Determine el comportamiento de la población para los valores restantes de f y P0.
- Implemente en Python un programa que modele el comportamiento de la población.

ANTECEDENTES

El modelo logístico de crecimiento poblacional, formulado inicialmente por Pierre François Verhulst en 1838, describe cómo una población crece de manera proporcional a su tamaño actual, pero también se ve limitada por factores ambientales. Este modelo ha sido ampliamente utilizado para estudiar sistemas biológicos y económicos, donde los recursos limitados influyen en el crecimiento. La ecuación utilizada en este estudio, $Pn+1=f\times Pn\times(1-Pn)$, es una forma discreta de este modelo que introduce una tasa de crecimiento f, la cual controla la dinámica poblacional. Dependiendo del valor de f, la población puede decaer, estabilizarse o entrar en comportamientos oscilatorios y caóticos. Este fenómeno es conocido como bifurcación y ha sido estudiado en la teoría del caos para entender las transiciones entre diferentes regímenes dinámicos.

DESARROLLO

A través del código implementado en Python, se modeló la dinámica poblacional utilizando la ecuación logística discreta para explorar cómo diferentes valores de la tasa de crecimiento f afectan el comportamiento a largo plazo de la población. Mediante simulaciones, se identificó que para $0 \leq f \leq 1$, la población eventualmente muere sin importar la población inicial P0. En el rango $1 < f < 2$, la población converge a un valor estable, lo que indica una bifurcación de primer orden. A valores más altos de f, el sistema comienza a mostrar comportamientos más complejos, como bifurcaciones adicionales y, eventualmente, caos. Se implementaron controles interactivos para f y P0, así como gráficos que ilustran la evolución temporal de la población y la tasa de bifurcación en función de f, lo que permitió una exploración visual detallada del comportamiento del sistema. Estas herramientas demostraron ser valiosas para comprender la rica dinámica que puede emerger incluso en sistemas tan simples como el modelo logístico discreto.

FUNDAMENTO MATEMÁTICO

Se parte de :

$$P_{n+1} = fPn(1 - Pn)$$

Donde

- Pn es una proporción de la población
- F es la constante de fertilidad

$P_{n+2} = Pn$, esto se da por un punto fijo donde la población es la misma

Tal que:

$$P = fP(1 - P)$$

$$P = fP - fP^2$$

$$P - fP = -fP^2$$

$$fP^2 - fP + P = 0$$

$$fP^2 - (f - 1)P = 0$$

$$P(fP - (f - 1) = 0$$

$$p = 0$$

$$fP - f + 1 = 0$$

$$P = \frac{f - 1}{f}$$

$$f = 1 \quad f = 2 \quad f = 1{,}5$$

$$P = 0 \quad P = 0{,}5 \quad P = 0,\widehat{33}$$

Para determinar las bifurcaciones $\boldsymbol{f \geq 0}; \boldsymbol{fe}[\mathbf{0}, \mathbf{4}]$

Partimos del análisis en un punto fijo

$$P = fP(1 - P)$$

Para establecer la función de iteración y así su estabilidad por el signo del valor absoluto de su derivada.

$$g(P) = fP(1-P), su\ derivada\ g'(P) = f(1-2P)$$

Ahora evaluamos en $P = \frac{f-1}{f}$, $P = 0$

Los valores partidos de f se identifican donde ocurre un cambio en el número de soluciones totales.

$$g'(0) = f\big(1-2(0)\big)$$

$$g'(0) = f$$

$$|g'(0)| < 1$$

Con, $|f| < 1$

$$P = 0, \qquad f < 1 \quad \wedge \quad f > 1$$

Estable Inestable

$$g'\left(\frac{f-1}{f}\right) = f\left(1-2\left(\frac{f-1}{f}\right)\right)$$

$$= f\left(1-\left(\frac{2f-2}{f}\right)\right)$$

$$= f\left(1-2\left(\frac{2f}{f}-\frac{2}{f}\right)\right)$$

$$= f\left(1-2+\frac{2}{f}\right)$$

$$= f\left(\frac{f-2f+2}{f}\right)$$

$$= f\left(\frac{-f+2}{f}\right)$$

$$g'\left(\frac{f-1}{f}\right) = -f+2$$

$$\left|g'\left(\frac{f-1}{f}\right)\right| < 1$$

$$|2-f| < 1$$

$$-1 < 2-f < 1$$

Con $-3 < -f < -1$

$$P = \frac{f-1}{f} \quad 1 < f < 3$$

Estable

$f > 3$ $el\ punto$, para:

$$P = \frac{f-1}{f}\ se\ vuelve\ inestable, lo\ que\ lleva\ a\ la\ aparición\ de\ bifurcaciones.$$

Para el diagrama de bifurcación:

Ecuación discreta

$$P_{n+1} = fPn(1 - Pn)$$

Consideramos que $P_{n+1} - Pn$ es un cambio infinitesimal en t, es decir $\Delta t \to 0$; por lo tanto

$$\frac{dP(t)}{dt} \approx \frac{P_{n+1} - Pn}{\Delta t}\ donde\ \Delta t\ es\ el\ intervalo\ de\ tiempo\ entre\ tn\ y\ t_{n+1}$$

Tq

$$P_{n+1} = fPn(1 - Pn)$$

$$P_{n+1} - Pn = fPn(1 - Pn) - Pn$$

$$\frac{dP(t)}{dt} = fP(t)\big(1 - P(t)\big) - P(t) \text{ donde f es la tasa de crecimiento}$$

Sin embargo, es necesario introducir un k, que limite el crecimiento máximo de la población:

$$\frac{dP(t)}{dt} = fP(t)\ \left(1 - \frac{P(t)}{K}\right) - P(t) \text{ y considerando que:}$$

K=1, volvemos a la expresión inicial

$$\frac{dP(t)}{dt} = fP(t)\big(1 - P(t)\big) - P(t)$$

Y bien, para determinar el diagrama de bifurcaciones en torno a f, usamos:

$$g(P) = fP(1 - P) - P \text{ y sus derivadas tal que:}$$

$$g^n(P) \text{ hasta que } f = 4$$

Con la gráfica obtenida podemos indicar la posición de f cumpliendo que:

$$P = g''(P) \text{ con un}$$

$$0 \leq f \leq 4$$

Mientras que en, $2.8 \leq f \leq 3.6$ cumple con:

$$P = g^8(P);$$

Adicional, un $P = g^8(P)$ con un f

$$2.8 \leq f \leq 3.6$$

Y finalmente, un $P_n = g^n(P)$, con un $n \in [500, 1500]$

Para indicar los P en sus

$$3.5 \leq f \leq 4$$

Para la resolución de esta y la creación de las gráficas se ocupó el Método de Euler:

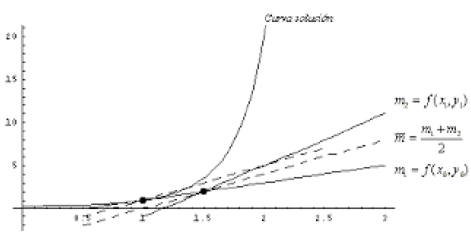

Figura 1. Representación Método de Euler

Llegando así a las aproximaciones de los puntos críticos donde se dan las bifurcaciones y obteniendo una gráfica similar.

Adicional a esto analizando la gráfica de bifurcación se puede decir que estas crecen en cada bifurcación en sentido de 2^n donde n es el número de bifurcación.

PSEUDOCÓDIGO / DIAGRAMA DE FLUJO

El pseudocódigo describe el flujo de una aplicación web interactiva desarrollada en Dash y Plotly para modelar y visualizar el comportamiento de una población en función de la tasa de fertilidad y la población inicial. La aplicación presenta tres gráficos principales: un gráfico Cobweb, una evolución temporal de la población, y un diagrama de bifurcación.

Primero, se inicializa la aplicación y se define el diseño de la interfaz, que incluye controles como deslizadores para ajustar la tasa de fertilidad y la población inicial, junto con botones para mostrar modales con instrucciones y detalles del proyecto. Las funciones centrales se encargan de calcular la evolución de la población usando un modelo logístico y de generar los gráficos interactivos. A través de callbacks, la aplicación actualiza dinámicamente los gráficos y el estado de los controles, reflejando los cambios introducidos por el usuario en tiempo real. Finalmente, el servidor de la aplicación se ejecuta, permitiendo la interacción con la interfaz a través de un navegador web.

1. **Inicializar la aplicación Dash:**
    - Usar el tema Bootstrap para los estilos.
2. **Definir el diseño de la aplicación:**
    - Contenedor principal (Container):
        - Alerta inicial con el título y las instrucciones.
        - Encabezado de la aplicación.
        - Fila que contiene los controles (deslizadores, botones) y los gráficos (Cobweb, Evolución de la Población, Bifurcación).
        - Almacenamiento de datos (Store) para los valores de los deslizadores f y P0.
3. **Definir la función population_model:**

- Inicializar un array para la población.
- Calcular la población para cada iteración usando la ecuación del mapa logístico.
- Devolver el array de la población.

4. **Definir la función p_futura:**
    - Calcular y devolver el valor futuro de la población en función de f.
5. **Definir la función cobweb_plot:**
    - Usar population_model para obtener los valores de población.
    - Crear el gráfico Cobweb:
        - Trazar la curva del mapa logístico.
        - Dibujar las líneas del cobweb para cada iteración.
    - Devolver la figura del gráfico Cobweb.
6. **Definir la función population_evolution:**
    - Usar population_model para obtener los valores de población.
    - Crear el gráfico de la evolución de la población.
    - Devolver la figura del gráfico de evolución.
7. **Definir la función bifurcation_diagram:**
    - Generar valores de f y calcular la población para múltiples iteraciones.
    - Crear el diagrama de bifurcación y devolver la figura del diagrama de bifurcación.

8. **Definir el callback update_graphs:**

    - Actualizar los gráficos Cobweb y de Evolución de la Población según los valores de f y P0.
    - Calcular el valor futuro de la población, devolver las figuras actualizadas y los valores de los deslizadores.

9. **Definir el callback update_store_f:**

    - Actualizar el valor almacenado de f cuando el deslizador cambia.

10. **Definir el callback update_store_p0:**

    - Actualizar el valor almacenado de P0 cuando el deslizador cambia.

11. **Definir el callback update_bifurcation_diagram:**

    - Actualizar el diagrama de bifurcación cuando cambia el valor almacenado de f.

12. **Definir el callback toggle_modal_instructions:**

    - Alternar la visibilidad del modal de instrucciones.

13. **Definir el callback toggle_modal_details:**

    - Alternar la visibilidad del modal de detalles.

14. **Definir el callback toggle_modal_info:**

    - Alternar la visibilidad del modal de información adicional.

15. **Ejecutar la aplicación Dash:**

    - Iniciar el servidor en modo de depuración.

Diagrama de flujo:

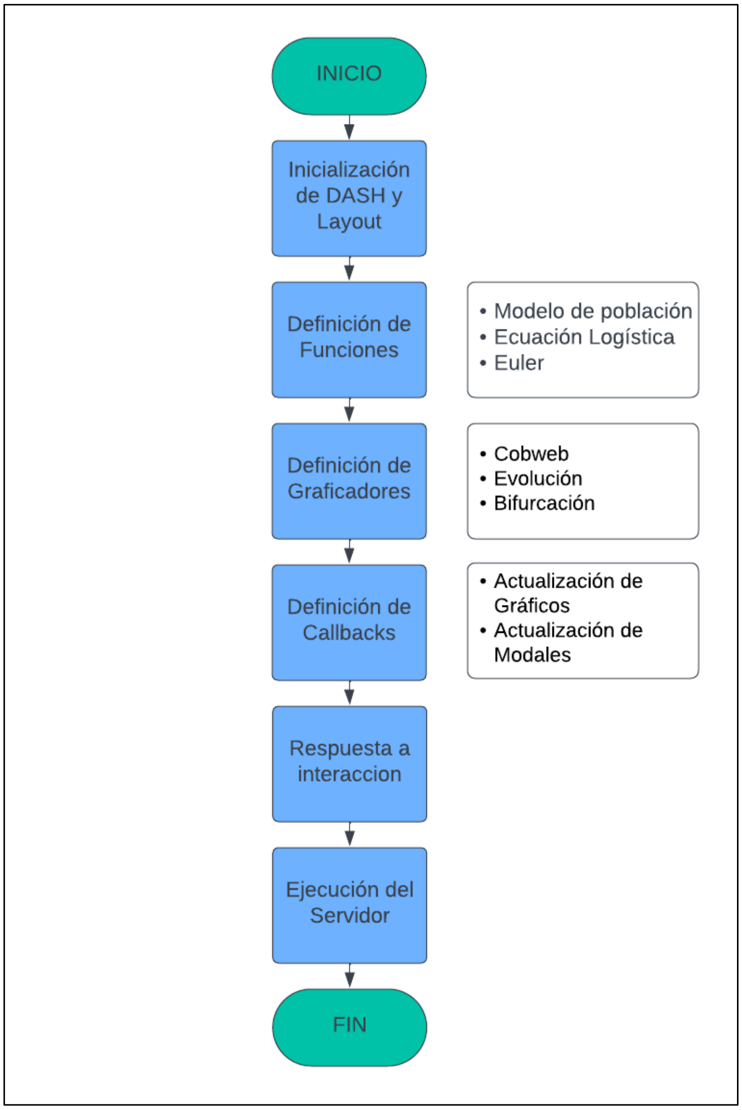

Figura 2. Diagrama de flujo.

CONSIDERACIONES IMPORTANTES DEL CÓDIGO

Estructura General:

1. **Imports y Configuración Inicial**:

    - Se importan las librerías necesarias: dash, dash_bootstrap_components, plotly.graph_objs, numpy, entre otros.
    - La aplicación se inicializa con Dash, utilizando un tema de Bootstrap para estilos adicionales.

2. **Diseño de la Interfaz**:

    - La interfaz de usuario se compone principalmente de componentes de Dash Bootstrap Components (dbc) y Dash Core Components (dcc).
    - **Contenedor Principal (dbc.Container)**:
        - Incluye una alerta con la introducción y las instrucciones de uso, un encabezado y varias filas (dbc.Row) que contienen columnas (dbc.Col) con deslizadores y gráficos.
        - **Controles**: Los deslizadores permiten ajustar la tasa de fertilidad (f) y la población inicial (P0).
        - **Modales**: Se incluyen modales que pueden abrirse para mostrar instrucciones, detalles del proyecto e información adicional.

Puntos clave:

- **Interactividad**: Los gráficos se actualizan dinámicamente en función de los valores seleccionados por el usuario.
- **Uso de Modales**: Los modales ofrecen información adicional sin recargar la página.

- **Gráficos Cobweb y Bifurcación**: Visualizan conceptos complejos como la estabilidad y el caos en un sistema de población iterativo.

El código se encuentra almacenado en el repositorio de GitHub y cuenta con los comentarios correspondientes de su descripción.

INTERFAZ GRÁFICA

1. Encabezado y Mensaje de Bienvenida:

- **Mensaje de Bienvenida:** Un mensaje introductorio dentro de un cuadro de alerta que incluye una breve explicación de la aplicación y las instrucciones de uso. Es posible cerrar este mensaje, y está ubicado en la parte superior de la interfaz.

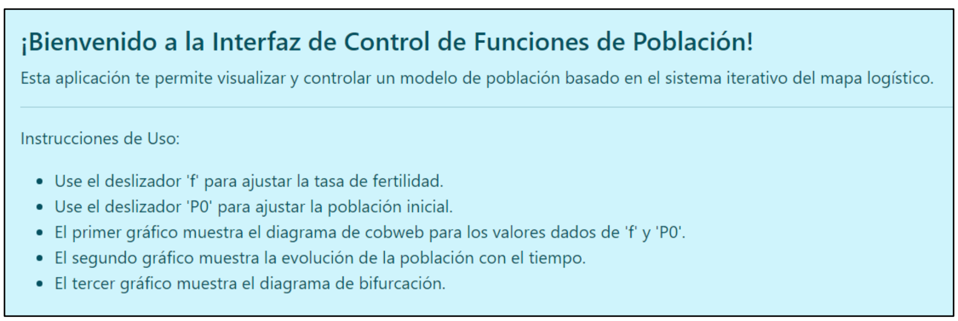

Figura 3. Mensaje de bienvenida e instrucciones.

2. Controles Interactivos:

- **Deslizador para la Tasa de Fertilidad (f):** Un deslizador horizontal que permite ajustar el valor de f en un rango de 0 a 4, con un paso de 0.01. Muestra el valor actual de f en un pequeño cuadro justo debajo del deslizador.

- **Deslizador para la Población Inicial (P0):** Otro deslizador horizontal para ajustar el valor de P0 en un rango de 0 a 1, con un paso de 0.01. También muestra el valor actual de P0.

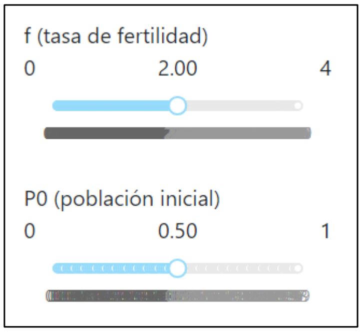

Figura 4. Sliders.

- **Botones de Modales:**
    - **Botón "INSTRUCCIONES":** Abre un modal con una guía paso a paso sobre cómo usar la aplicación.
    - **Botón "DETALLES":** Abre un modal con información sobre el proyecto, describiendo su propósito y los gráficos que se generan.
    - **Botón "ACERCA DE":** Abre un modal con información adicional sobre el proyecto, como los nombres de los integrantes y la institución educativa.

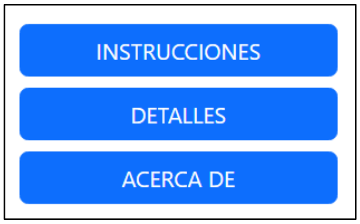

Figura 5. Botones de información.

3. Visualización de Gráficos:

- **Gráfico Cobweb:** Una visualización que muestra la dinámica iterativa del modelo de población con un diagrama cobweb, que refleja cómo la población evoluciona en cada iteración dependiendo de los valores seleccionados para f y P0.

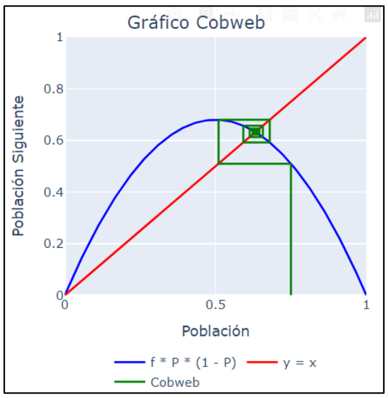

Figura 6. Grafico Cobweb.

- **Gráfico de Evolución de la Población:** Un gráfico de líneas que muestra cómo cambia la población a lo largo del tiempo, basado en los valores actuales de f y P0.

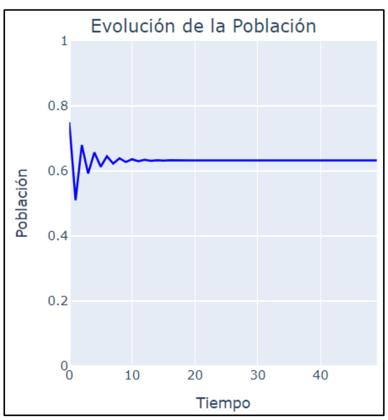

Figura 7. Gráfico de la evolución de la Población en el tiempo.

- **Diagrama de Bifurcación:** Un gráfico que visualiza cómo la población se bifurca (o diverge) a medida que varía la tasa de fertilidad f, manteniendo constante el valor de P0.

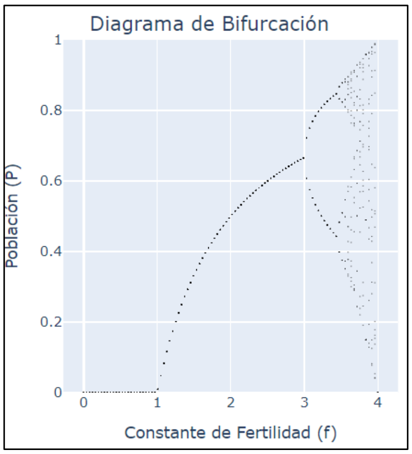

Figura 8. Diagrama de Bifurcación.

- **Gráfico de Matplotlib convertido a Plotly:** Un gráfico generado mediante Matplotlib que muestra la solución de una EDO relacionada con la dinámica de población. Este gráfico no depende de los sliders y muestra múltiples curvas que representan distintas combinaciones de parámetros iniciales.

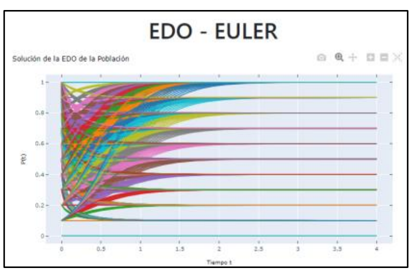

Figura 9. Gráfico de poblaciones.

Con formato: Normal, Izquierda

4. Almacenamiento de Estado:

- **dcc.Store Componentes:** Se utilizan para almacenar los valores de f y P0 entre callbacks, permitiendo que la interfaz se mantenga reactiva y actualice los gráficos en función de estos valores almacenados.

5. Interfaz Completa

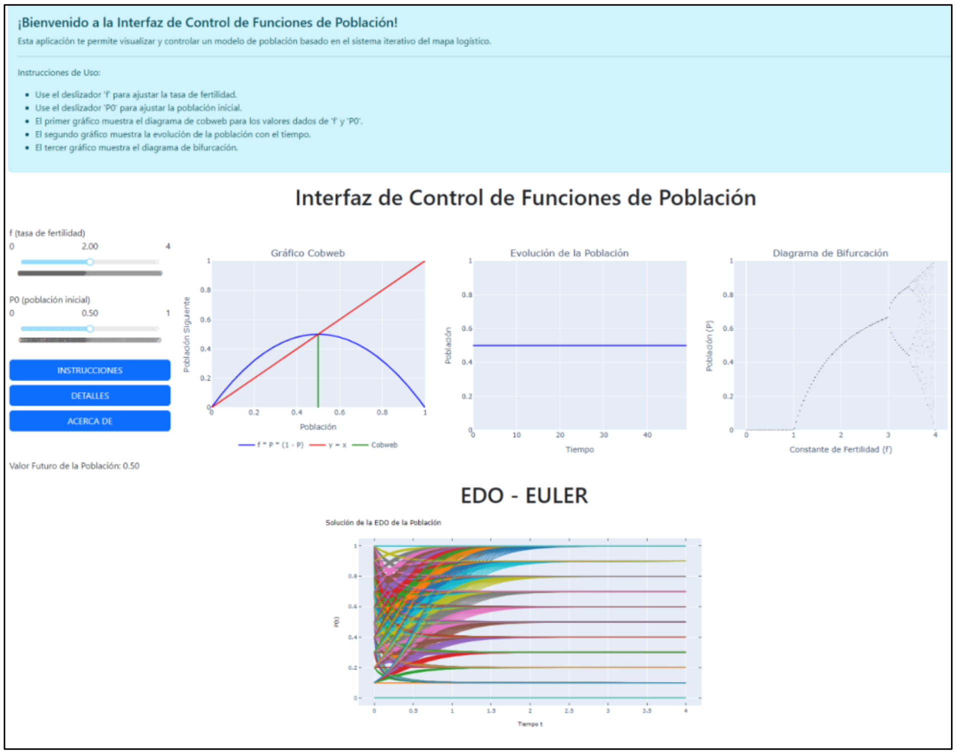

Figura 10. Interfaz gráfica completa.

CONCLUSIONES Y RECOMENDACIONES

- Para $0 \leq f \leq 1$, la población muere, lo que se asocia con 0 bifurcaciones.
- Para $1 < f < 2$, la población converge a un valor fijo, indicando 1 bifurcación.
- Para $2 < f < 2.5$, se observan 2 bifurcaciones, donde la población oscila entre dos valores.
- A medida que f aumenta más allá de 2.5, el número de bifurcaciones incrementa, y se pueden observar comportamientos caóticos a valores mayores de f.
- **Para f bajo:** La población decae a 0 (muerte).
- **Para f intermedio:** La población puede estabilizarse o bifurcarse en un ciclo limitado.
- **Para f alto:** La población puede exhibir comportamientos caóticos donde no se establece un patrón claro.
- Implementar sliders para f y P0 permite a los usuarios interactuar con el modelo y observar cómo pequeños cambios en estos parámetros afectan el comportamiento de la población. Es una herramienta valiosa para visualizar bifurcaciones y puntos fijos.
- La gráfica de la evolución de la población a lo largo del tiempo muestra claramente cómo la población puede estabilizarse, oscilar, o mostrar comportamientos caóticos. Esto ayuda a entender la dinámica a largo plazo del sistema.
- La gráfica de bifurcaciones ilustra el fenómeno del "doble de periodo" y el caos que surge a medida que f aumenta. Es una representación visual poderosa que ayuda a identificar transiciones entre diferentes tipos de comportamiento poblacional.

REFERENCIAS

- Libretexts. (2022, 2 noviembre). *4.4: Crecimiento Natural y Crecimiento Logístico*. LibreTexts Español. https://espanol.libretexts.org/Matematicas/Matematicas_Aplicadas/Matematicas_universitarias_para_la_vida_cotidiana_(Inigo_et_al.)/04%3A_Crecimiento/4.04%3A_Crecimiento_Natural_y_Crecimiento_Log%C3%ADstico
- Aliet. (2018, 2 octubre). Dinámica de Sistemas usando PySD - aliet - Medium. *Medium*. https://medium.com/@aliet_7/din%C3%A1mica-de-sistemas-usando-pysd-1ddaaa9daac4
- Carakenio. (2023, 22 septiembre). *Resolver ecuaciones diferenciales en tiempo discreto con Matlab*. Dademuchconnection. https://dademuchconnection.wordpress.com/2020/10/21/resolver-ecuaciones-diferenciales-en-tiempo-discreto-con-matlab/
- Libretexts. (2022b, noviembre 2). *11.1: Puntos fijos y estabilidad*. LibreTexts Español. https://espanol.libretexts.org/Matematicas/Ecuaciones_diferenciales/%C3%81lgebra_Lineal_Aplicada_y_Ecuaciones_Diferenciales_(Chasnov)/03%3A_II._Ecuaciones_diferenciales/11%3A_Ecuaciones_diferenciales_no_lineales/11.01%3A_Puntos_Fijos_y_Estabilidad
- *Logistic Map, Euler y Runge-Kutta Method and Lotka-Volterra Equations*. (2013, 23 septiembre). [Diapositivas]. Department of Mathematical Sciences Seoul National University. https://www.math.snu.ac.kr/~syha/appliedpde.pdf